

University of London
Imperial College of Science, Technology and Medicine
Department of Computing

A Probabilistic Approach to Tracking Deformable Patches for Image-Guided Surgery

Osemwaro Pedro

Submitted in part fulfilment of the requirements for the degree of
Doctor of Philosophy in Computing of the University of London and
the Diploma of Imperial College, January 2011

Abstract

Totally-Endoscopic Coronary Artery Bypass operations, which should ideally be entirely performed with a robotic surgical assistant, currently suffer from a high rate of conversion to more traditional procedures, partly due to the difficulty of identifying the coronary artery that is to be operated upon. One solution to this problem is to guide the surgeon by superimposing models of patients' hearts onto the images provided by the robot. Motivated by the possibility of using motion information to (partially) constrain the registration of the model to the images, this thesis focuses on methods of estimating the motion of salient patches on the myocardial surface.

We begin by introducing an importance sampling algorithm for hypothesising affine patch transformations in a particle filtering framework. The algorithm minimises uncertainty by multiplicatively combining information from multiple patch subregions. We devise a method for handling missing information based on empirical evidence that suggests that certain importance sampling probability ratios grow empirically with the number of subregions.

We then describe methods for calculating the dissimilarity between image regions whilst taking into account specular reflections and illumination changes. We achieve insensitivity to these effects by explicitly removing illumination changes, tentatively masking out specular reflections, and ignoring pixel differences that exceed a percentile of a weighted pixel difference distribution.

Next, we investigate myocardial deformation sequence models, and propose nesting a PCA model of the static deformations within a periodic B-spline-based PCA model of the deformation sequences. We use this model to simulate data sets that we can use to approximate maximum likelihood estimates of some parameters of the particle filter components, and we describe a way of testing whether or not particles have entered low-probability states in which they cease to contribute useful information.

Acknowledgements

First and foremost, I would like to thank my supervisor Daniel Rueckert for his guidance, patience, willingness to give me complete freedom to explore my own ideas, and for not giving up on me despite the numerous dead ends I encountered. I am also grateful to: my second supervisor Eddie Edwards, particularly for assisting me in acquiring the intraoperative data I used, and for not giving up on me despite my occasional late arrivals at the operating theatre; the EPSRC for grant EP/C523008/1 that funded the project that my work was a part of; and the Department of Computing at Imperial College London for granting me the DTA award that funded my studies.

I am also inexpressibly grateful to the many wonderful friends I have made over this four year journey. In particular, I would like to thank: Paul for his mathematical advice, his music/comedy recommendations, and for not giving up on me despite me bombarding him with emails on topics as diverse as differential geometry and “quantum-mechanical equestrianism”; Delphine, Ute, Fiona and Karim for the many concerts, restaurants, trips etc. that we have enjoyed together, and Delphine especially for all the fun music-making sessions and for not giving up on me despite my crippling stage fright that made many of our chamber recitals less splendid than they could have been; Dong Ping for her wonderful assortment of Chinese teas, for sneaking into lecture theatres with me to covertly watch films on the weekends, for regularly pummelling my back with her unexpectedly powerful fists, and for the many emails she sent me that consisted of nothing but a single question mark; Alexei and Uri for providing me with countless, much-needed “short breaks” that, with probability 1, turned into long evenings of unashamedly ultra-geeky humour; Arnaud and Ralph for all the jazz jam sessions and for not giving up on me despite my poor style of improvisation; Maria for our jazz singing sessions; Claire for her enthusiastic championing of Linux and the Liberal Democrats (not to suggest that I am particularly partial towards the latter); Kanwal for her multilingual birthday wishes; and the Imperial College Symphony Orchestra for giving me the opportunity to spend seven years playing so much of the most glorious music I know.

However great the magnitude of the inexpressibility of my gratitude to my friends may be, my gratitude to my family must, at the very least, be a point of even greater magnitude in the Banach space of ineffable things, if not an entire subspace of such points covering the uncountably infinite ways in which I cherish them. This is a corollary to the following lemmas

(I leave the proof as an exercise for the reader): 1) were it not for my parents, I would not be; 2) were it not for my family, I would not be the person that I am; 3) were it not for my family's inexhaustible love, encouragement and laughter, I would not have embarked upon this long academic voyage in the first place.

And finally, I would like to thank the makers of the fine Colombian coffees that I have been consuming in unhealthily large doses over the past few months. Were it not for their efforts, I would probably have less of a headache now, but also less of a thesis.

Dedication

To the many wonderful teachers I have had

“Statistics are like bikinis. What they reveal is suggestive, but what they conceal is vital.”

Aaron Levenstein

Contents

Abstract	3
Acknowledgements	5
1 Introduction	31
1.1 Motivation and Objectives	31
1.2 Contributions	35
1.3 Statement of Originality	36
2 Related Work	37
3 A Product-of-Experts Approach to Importance Sampling	41
3.1 Synopsis	41
3.2 Affine Patch Tracking with a Particle Filter	42
3.2.1 Probabilistic Approximations	43
3.3 Product-of-Local-Likelihoods Importance Sampling	48
3.3.1 Local Experts	52
3.3.2 Discrete, Bounded-Displacement Energy Functions	54
3.3.3 A First-Order Approximation of $\ell_t^{[j]}$	55

3.3.4	Local Likelihood Energy Problems	59
3.4	Results: Exponential Likelihood Ratio Growth Study	71
3.5	Conclusion	79
4	Likelihood Evaluation	82
4.1	Synopsis	82
4.2	Robust Image Region Comparison	83
4.3	Lightness Normalisation Filtering	92
4.3.1	Illumination Model	93
4.3.2	Related Work	95
4.3.3	Local Lightness Normalisation	97
4.3.4	Parametric Surface Fitting	101
4.4	Specular Highlight Masking	104
4.5	Results	106
4.5.1	Image Filtering	106
4.5.2	Comparative Tests	109
4.6	Conclusion	112
	Issues to be Addressed in the Remainder of this Thesis	114
5	Deformation Modelling	116
5.1	Synopsis	116
5.2	Dense Deformation Field Estimation	117
5.3	Deformation Mode Modelling	125

5.3.1	Deformation Field Inversion	127
5.3.2	Affine Point Set Registration	130
5.3.3	Sampling Phase Offsets and Canonical Grid Similarity Transformations .	133
5.3.4	PCA-Based Deformation Mode Models	138
5.3.5	Polar Coordinate PCA-Based Deformation Mode Models	142
5.4	Deformation Trajectory Mode Modelling	146
5.4.1	Deformation Trajectory Models	147
5.4.2	B-Spline Inner Product Spaces	149
5.4.3	A B-Spline-Based Gaussian DTM	153
5.5	Results	159
5.5.1	Deformation Mode Models	165
5.5.2	Deformation Trajectory Mode Models	169
5.6	Conclusion	175
6	Towards a Strategy for Coping with Particle Loss	176
6.1	Synopsis	176
6.2	Introduction	177
6.3	Detecting Particle Loss	180
6.3.1	Likelihood-Ratio-Based Particle Loss Inference	181
6.4	Parameterising the Particle Loss Test	185
6.4.1	DTM-Simulation-Based Sample Set Creation	185
6.4.2	Simulation Details	189
6.4.3	Stretched Exponential Likelihood Parameterisation	196

6.4.4	Selecting the Test Threshold	202
6.5	Some Methods for Restoring Lost Particles	206
6.6	Results	210
6.6.1	Loss Test Results	212
6.6.2	Particle Filter Accuracy	220
6.6.3	Comparative Tests	221
6.7	Conclusion	228
7	Conclusion	229
7.1	Summary of Thesis Achievements	229
7.2	Future Work	231
A	Notational Conventions	234
B	The After Math	236
B.1	Calculating the Decomposition $\mathbf{A} = \mathbf{RSK}$	236
B.2	The Existence and Realness of 2×2 Matrix Logarithms	238
B.3	Calculating the Discretisation Resolution Parameters	239
B.3.1	The Rotational Discretisation Parameter δ_ϕ	240
B.3.2	The Skewing Discretisation Parameters δ_{κ_i}	241
B.3.3	The Scaling Discretisation Parameters δ_s	242
B.4	Maximising the Importance Sampling Distribution Entropy	243
B.5	Maximising a Particle Set's Entropy	245
C	Algorithms	247

List of Tables

3.1	The values of the particle filter parameters that we manually defined. The unit p stand for “pixels”. Unless otherwise stated, we use the same values for all tests in this thesis. <i>Note: \mathcal{E} is a dissimilarity function that we will use to define Δ_{ℓ} and Δ_L in the next chapter.</i>	72
3.2	The percentiles of the number of subregions that defined the particles’ sampling distributions.	72
5.1	The parameters we used for each type of canonical grid and their degrees of freedom (DoF). The parameter symbols are the same as those we used in section 5.3.3. As mentioned earlier, the innermost rings of the unconstrained quasi-circular grids have just 2 DoF, because they are degenerate. However the innermost rings of the constrained quasi-circular grids (including the polar grids) have 0 DoF, as their vertices always lie at the origin.	161
5.2	This table shows the ranges of diameters and maximum edge lengths of the source grids. The maximum edge length of a grid is the maximum distance between neighbouring vertices. The diameters of the square grids are calculated as the distances between diagonally-opposite vertices. Their range of side lengths was [100, 180].	161
5.3	This table shows the percentages of source grids that contained at least as many landmarks as are indicated in the table’s body.	161
5.4	Landmark sequences: summary statistics.	162

5.5	Sampled values of M for which the 98 th percentiles of the “Square” and “Circular, Cartesian” landmark reconstruction errors fell within the 7 pixel limit for 180608.HS.	174
5.6	Sampled values of M for which the 95 th percentiles of the “Square” and “Circular, Cartesian” landmark reconstruction errors fell within the 7 pixel limit for 230108.FO1.	174
6.1	This table summarises the parameters that we used to define the distributions of the perturbations that we applied when constructing the D_* set of patch/subregion differences. As always, the unit p stands for pixels.	211

List of Figures

1.1	The da Vinci system (a) and a close-up of its endoscope (b). The two large circles at the bottom of the endoscope are the stereo cameras, and the smaller circle above them is a light. <i>The copyright of these images is owned by Intuitive Surgical [51].</i>	33
1.2	This figure shows some of the ways in which images from monoscopic and/or stereoscopic cameras can be processed in order to extract shape information for augmented reality. Shape-from-stereo methods involve exploiting the disparities between corresponding points in a set of simultaneously captured images in order to estimate depth. Shape-from-shading methods use lighting models to estimate depth from changes in surface intensity. Motion information can be used to estimate shape directly, or to place temporal consistency constraints on methods that normally generate independent shape estimates from one time step to the next. <i>The top-right image is taken from [80].</i>	33
3.1	An example of the non-uniqueness of the decomposition in eq. (3.2), in which the red square's state can be expressed as a transformation of the blue square's with two different instantiations of θ	44

- 3.2 (a) The bent purple line at the bottom moves to the bent line at the top. Arrows between encircled regions indicate high-likelihood circle translations; each circle has high uncertainty in the direction parallel to the line segment it lies on. Pairs of circles not joined by arrows have low translation likelihoods. Taking each region's displacement likelihood as an expert, a Product-of-Experts model gives high probability to the displacement indicated with solid red arrows, and low probability to all other displacements. (b) The pointwise product of a blue circle's likelihood function and a green circle's likelihood function annihilates uncertainty, producing a delta function. 50
- 3.3 (a) The crosses are grid vertices in G . The coloured crosses denote grid vertices lying within the red patch. The blue cross and square denote a subregion of the patch. The patch's vertices lie within the shaded grid squares. (b) After an affine transformation in which the patch's vertices remain within the shaded grid squares, only a few red grid vertices leave the patch (the ones with black circles around them), and only a few black ones enter it (the ones with red circles around them). 53
- 3.4 A 30° clockwise subregion rotation about the cross in the centre can be approximated by rotating the subregion centres, but preserving the subregions' alignment with the image boundaries. 54
- 3.5 (a) A patch on the myocardium (green square), and one of its subregions $S(\mathbf{g}_{t-1}^{[.]})$ (blue square). The time t image is omitted, but it is mostly the result of a small displacement of the shown time $t - 1$ image downwards and to the right. (b) A subregion displacement likelihood function for $S(\mathbf{g}_{t-1}^{[.]})$'s displacements, defined as $e^{-\psi(\cdot; \mathbf{g}_{t-1}^{[.]})}$. Light regions denote high displacement likelihood (low energy), and dark regions denote low likelihood (high energy). 56
- 3.6 Dashed parallelograms denote patches defined by time $t - 1$ particles. Patches sharing a hash key (defined by the shaded squares that patch vertices lie in) are grouped together by colour, and the group averages are represented by the solid parallelograms. 60

- 3.7 The graphs in this figure represent two possible realisations of three energy functions. In (a), the energy function at the bottom is undefined everywhere in its domain (as indicated by the red crosses), and in (b) each energy function is partially undefined. In both cases, the total number of defined energy terms for each domain state x over the three functions is two. Furthermore, the sum of the defined energies over all three functions is the same for each corresponding state x in the two figures (the colours of the bars indicate that (b) is simply the result of swapping the values of the energy functions in (a)). Thus under eq. (3.26), both sets of energy functions would define the same conditional distribution $k^{-1}\tilde{f}(x)$ if we simply omitted the undefined energies. 61
- 3.8 The 3×3 numbered time $t - 1$ subregions either remain in their current position or move to one of the 8 positions surrounding them, as shown in the time t diagram. The pale blue blob indicates a masked out region of the time t image, which destroys all of the likelihood information for subregions (1), (4) and (7), as indicated by the red squares in the displacement likelihood grids on the right. In these grids, the black loop around the rightmost 6 squares indicates displacements for which at least one likelihood value is defined, and the squares indicated in purple highlight the likelihoods for displacement $(1, 1)^T$ 67
- 3.9 5 frames from the 20/19 frames over which we tracked four patches. 74
- 3.10 An illustration of the margin α' that separates the partial means $\bar{\psi}_{x,i:n}$ from the partial means $\bar{\psi}_{x_{\tilde{f}}^{\downarrow},i:n}$, for $i = i_{\tilde{f}}(x, u)$ (for some arbitrary u) and $n = \min(n_{\tilde{f}}(x), n_{\tilde{f}}(x_{\tilde{f}}^{\downarrow}))$. The latter partial means are indicated by the green markers on the left (the dot marks the $n_{\tilde{f}}(x_{\tilde{f}}^{\downarrow})^{\text{th}}$), and the former are indicated by the red markers on the right. The error bars on the two groups indicate the regions about the dots within which the other partial means must lie, as dictated by the 100th percentile curve of the ARE histograms. Increasing the value of u decreases the width of these bounds, and hence increases the margin α' 78

- 3.11 (a), (c), (i), (k) Histograms of the joint distributions of the pairs from eq. (3.56), for the 230108.FO1 video. In all subfigures, the x axis denotes values of $\frac{i}{n_{\tilde{f}}(x)}$. The y axes have been partitioned into 3000 bins, and the x axes into 100; we do this with all histograms in the rest of this section. The colour key follows a logarithmic scale, and no samples fell into the black regions. (b), (d), (j), (l) Histograms of the joint distributions of the pairs from eq. (3.59). (e)-(h), (m)-(p) The 1st, 10th, 25th, 50th, 75th, 90th and 99th percentiles of the histograms above them, under the condition that $\frac{i}{n_{\tilde{f}}(x)}$ lies in a specific bin. 80
- 3.12 The figures here have a similar interpretation to those in figure 3.11, except that these figures are for the 180608.HS video. 81
- 4.1 Three images showing the kinds of changes in the appearance of an image region that the dissimilarity functions must be able to cope with. (a) The original region. (b) An illumination change. (c) Occlusion by specular highlights. 83
- 4.2 This figure illustrates the relationship between our definition of h (depicted by the blue horizontal line segments), the bin boundaries (filled circles indicate closed bounds, unfilled represent open bounds), and the CDF H_p (the green dot-dashed line segments). In this example, the mass of h over each bin and the gradient of H_p over each bin decrease by a factor of half from left to right. We approximate $\varrho[H_p^*, H_\delta]$ as $\approx \varrho[H_p, H_\delta]$, which is the area above H_p and below the line $y = 1$ 87
- 4.3 (a) The interactions between light and an inhomogeneous material. (b) The photometric angles used in the Dichromatic Reflection Model. *Image (a) taken, and image (b) adapted, from [97].* 95
- 4.4 (a) A comparison of $C'(Y)$ and $a \log(Y) + b$. (b) A plot of the absolute relative error $\frac{|C'(Y) - a \log(Y) - b|}{C'(Y)}$ 99
- 4.5 An illustration of the manner in which we arrange the squares over which we fit the surfaces. We define the overall value of the surface in regions covered by two squares, such as the red rectangle, by cubic interpolation. In regions covered by four squares, such as the green square, we use bicubic interpolation. 101

- 4.6 (a) An image from our 230108.FO1 sequence. (b) The result of applying Tan and Ikeuchi’s method [107] to (a) to remove the specular highlights, as computed by their source code. (c) An intraoperative image used by Stoyanov and Yang. (d) Stoyanov and Yang’s result after removing specular highlights from (c). Images (c) and (d) taken from [104]. 104
- 4.7 Examples of our lightness normalisation and specular highlight masking results for frames 1 and 47 of the 230108.FO1 sequence. For lightness normalisation, we used 60-pixel-wide planar surfaces with 30-pixel-wide overlaps (the image dimensions are 768×576). To facilitate comparison, we have added the mean lightness of frame 0 to all lightness normalised images. (a), (g) The original images from frames 1 and 47. (b), (h) The original L^* channels. (c), (i) Specular highlight masks – highlights shown in cyan. (d), (j) The result of applying lightness normalisation to the L^* channels and converting the images back to RGB space. (e), (k) The lightness normalised L^* channels. (f), (l) All of the S' surfaces used in lightness normalisation, rendered side by side. 107
- 4.8 Examples of our lightness normalisation and specular highlight masking results for frames 38 and 62 of the 180608.HS sequence. All of the comments in the previous figure’s caption apply, except that we used circular parabolic surfaces in the lightness normalising filters this time. 108
- 4.9 These figures show percentiles of the weighted error distributions that resulted from tracking 14 patches with our proposed method (leftmost column of each group), from tracking the same patches without lightness normalisation (middle columns), and from tracking the patches using the weighted sum-of-squared differences rather than \mathcal{E} (rightmost columns). The IDs identify each of the patches that we tracked in each video. See figure 6.8 on page 212 to see what the patches looked like. 111

- 5.1 An example of the trajectories followed by 5 manually-tracked points over a cardiac cycle, with units in pixels. The circular markers denote the locations of the points at 5-frame intervals. Each trajectory begins at the top-left marker. The range of displacements of the points, relative to their initial positions, is up to 300 pixels horizontally and 200 pixels vertically. The apparent lack of periodicity is due to the effects of respiratory motion. 118
- 5.2 (a) This figure shows the median Euclidean error between the true landmark positions and the Thin-Plate Spline estimates of their positions in each frame for various values of λ . (b) This figure shows the maximum Euclidean fitting errors. In this case, the largest value of λ for which the errors are ≤ 5 pixels is $\lambda \approx 2500$ 122
- 5.3 (a) The reference image from the 180608.HS sequence with manually-placed landmarks, with units in pixels. Lightness normalisation has not been applied. (b) The image from the sequence that appeared to have the greatest amount of residual motion after estimating the deformation and backprojecting. (c) The estimated deformation, with the landmarks from image (b) shown. The red pluses show the true landmark positions, and the blue crosses show the deformation field's estimates of their positions (d) Close up of the reference image. The indicated convex hull is the region of interest. (e) The result of backprojecting image (b). (f) The difference between the reference image and the backprojected image. Note that difference images do not highlight deformation estimation errors in the homogeneous regions. 123
- 5.4 (a) The time t_r regular grid G is deformed by $\boldsymbol{\nu}_t$. Each coloured quadrilateral $\boldsymbol{\nu}_t(S)$ in the deformed grid corresponding to a grid square S of G has a bounding box indicated with dashed lines and is added to $H(S')$, for each square S' of G' shaded in a similar colour to $\boldsymbol{\nu}_t(S)$. (b) The bilinear interpolation coefficient α , which is used in the calculation of the position of $\boldsymbol{\nu}_t^{-1}(\mathbf{p})$ in the grid square of G that is mapped to $\square abdc$, makes $\mathbf{a} + \alpha \overrightarrow{ac}$, \mathbf{p} and $\mathbf{b} + \alpha \overrightarrow{bd}$ collinear. 129

- 5.5 (a) The 10×10 square canonical grid that we used in our experiments, with $2 \times 10 \times 10 = 200$ degrees of freedom. (b) A 6 ring, 20 sector quasi-circular canonical grid that we used. The 20 points of the degenerate innermost ring will always be collocated after any transformation, hence the grid has $2 \times (5 \times 20 + 1) = 202$ degrees of freedom. The two grids have equal area and a similar number of degrees of freedom. 135
- 5.6 (a) A point \mathbf{p}' uniformly sampled within parallelogram $\square \mathbf{wxy}$ can be linearly mapped to a point \mathbf{p} in triangle $\triangle \mathbf{wxy}$, giving a uniformly-distributed sample within that triangle. (b) After selecting \mathbf{o} , \mathbf{p} and Φ , the smallest distance u from \mathbf{p} to $\mathcal{S}_\mathbf{o}$'s boundary determines the upper bound on the scale factor \mathbf{a} 136
- 5.7 The ring 3 vertex of the spoke of g denoted by the solid black line is at a distance l from the origin, and it rotates away from the corresponding mean directional vector (shown in grey) by δ_ϕ radians. 145
- 5.8 This figure shows an example of the result of fitting a B-spline to a quasi-periodic deformation vector subtrajectory. The crosses mark the subtrajectories of the first two principal components of the deformation mode model and of two of the matrix logarithm entries, and the solid lines of matching colours are the maximum likelihood B-splines that model them. The vertical red dashed line marks the time point $\frac{\mathbf{t}-\mathbf{t}_r}{T}$, and the vertical black solid line behind it marks the candidate phase offset calculated over the B-spline using a method that we will describe in the next chapter. The data comes from a test based on square canonical grids over the 230108.FO1 data set that our results in the next section are partly based on. The parameters we used were $c = 23$, $d = 3$, $m = 40$ 158
- 5.9 Cumulative distributions of deformation field edge lengths (in pixel units). . . . 162
- 5.10 Cumulative distributions of deformation field landmark reconstruction errors (in pixel units). 162
- 5.11 Six deformation fields for video 180608.HS. The red pluses show the true landmark positions, and the blue crosses show the deformation fields' estimates of their positions. 163

5.12	Six deformation grids for video 230108.FO1. As before, the red pluses show the true landmark positions, and the blue crosses show the deformation fields' estimates of their positions.	164
5.13	Landmark reconstruction error percentiles for two deformation mode models trained and tested on 230108.FO1.	166
5.14	Deformation mode model landmark reconstruction error percentile ranges (in pixel units). See text for description.	167
5.15	These figures show the mean and first 10 modes of the deformation mode models for all 5 grid types. The top row contains the means, and the remaining rows show the modes, with eigenvalues decreasing from top to bottom. To highlight the kinds of deformations that the modes represent, we have scaled them all by nine standard deviations.	168
5.16	DTM landmark reconstruction error percentiles for 230108.FO1 under different numbers of B-spline control points (in pixel units). See text for description. . . .	171
5.17	DTM landmark reconstruction error percentiles for 180608.HS under different numbers of B-spline control points (in pixel units). See text for description. . . .	172
5.18	Eight samples from left to right of the mean trajectories and first nine modes of the square grid DTMs. To highlight the kinds of deformation sequence that the modes represent, we have scaled them by six standard deviations.	173
6.1	The encircled regions of (a) and (b) are an example of the kind of appearance change in which the colour of a feature (the near-vertical vessel section in (a)) fades away. (c) and (d) show some other changes of colour/texture that the DTM cannot generate. In some regions, colours seem to flow into each other, but in others new colours seem to appear from nowhere. Constructing D_L by perturbing $\Theta_t^{(k)}$ can account for the former of these occurrences to an extent, but it cannot account for the latter. All four images were taken from the 230108.FO1 sequence, after lightness normalisation. (a) and (c) are from frame 11 of the sequence, and (b) and (d) are from frame 21.	188

- 6.2 Some frames from a 50 frame deformation sequence sampled from the DTM that we constructed for the 180608.HS sequence and applied to the region of the first image surrounding the initial feature state shown in (a). For reference, we have rendered a regular grid with 24-pixel-wide squares over each image. 191
- 6.3 Stretched exponential distributions for various values of α and γ 197
- 6.4 These figures show histograms for the D_L and $D_{\ell'}$ sample sets generated for some patches from the 230108.FO1 sequence and the 180608.HS sequence. The rightmost number underneath each figure is an identifier for the patches – figure 6.8 shows what these patches looked like. The stretched exponential distributions that maximise the conditional likelihood of each sample set are shown as (truncated) solid red lines. 199
- 6.5 The three possible states of $B_G(b)$ relative to $cF_G(b)$, for some bin b and some value of c 204
- 6.6 (a) The red curve shows $cF_G(\delta)$ as a function of δ for feature 35 of the 180608.HS sequence, using optimal values for c , α_L and γ_L , calculated as described in this section and the previous section. The discontinuous blue line segments show $B_G(\delta)$ for the same feature. (b) A graph of $f(c)$ for the same feature. f has numerous plateaus, which arise due to the fact that cF_G does not always intersect B_G 205
- 6.7 The green and red particles are to compare the blue time $t - 1$ subregion to time t subregions using image information from time 0. As the time $t - 1$ states of the particles are very different, the subregions of the first frame from which they will retrieve information (indicated by the small green and red squares) are also very different. 207
- 6.8 These figures show the ID numbers that we used to refer to each of the patches that we tracked. 212
- 6.9 Screenshots of our particle filter’s performance over the 230108.FO1 video, taken at 5 frame intervals. The quadrilaterals show the weighted means of the particles, and the mostly-blue crosses near their centres show the positions of the particles. 213

- 6.10 Screenshots of our particle filter's performance over the 180608.HS video, taken at 5 frame intervals. The quadrilaterals show the weighted means of the particles, and the mostly-blue crosses near their centres show the positions of the particles. 214
- 6.11 These graphs show the trajectories of the weighted means of the particles in each video. The circular markers denote the positions of the weighted means at 5-frame intervals. 215
- 6.12 (a), (b) These figures show the changes in the distribution of the number of lost and non-lost particles over time. The numbers are always multiples of 50, which is the number of particles we used. This is because we resampled the lost particles whenever possible, meaning that either all of a patch's particles were lost after processing each frame, or none were. (c), (d) These figures show the numbers of lost and non-lost particles per patch, accumulated over all frames that they were tracked over. Again, the values are all multiples of 50. 216
- 6.13 (a), (b) Error percentiles per frame after partitioning particles according to whether or not they were lost. (c), (d) Error percentiles per patch after partitioning particles according to whether or not they were lost. A pair of dashed vertical lines is given for each patch. The lines on the left mark the percentiles for lost particles, and the lines on the right mark the percentiles for non-lost particles. Patch 69 in 230108.FO1 only has one vertical line, because the loss test erroneously labelled all of its particles as lost. 217
- 6.14 These figures show weighted percentiles of the overall patch errors and displacement patch errors. 222
- 6.15 These figures show weighted percentiles of the rotational and scaling patch errors. 223
- 6.16 These figures show weighted percentiles of the skewing patch errors. 224

6.17	These figures show some percentiles of the weighted displacement error distributions for each patch in the tests in which we compared the particle filter's performance when estimating full affine transformations to its performance when only estimating patch displacement. The left-hand column of each patch's results gives the error percentiles from the tests in which we estimated full affine transformations, and the right-hand column gives the displacement-only percentiles.	226
6.18	For each frame of the two videos, these figures show the percentiles of the distributions over the 14 patches of the pyramidal KLT tracker's (p-KLT, [14]) displacement errors and the errors in our particle filter's weighted mean estimates of the patch displacements. The dashed lines show the p-KLT tracker's results, and the solid lines show our particle filter's results. In the figures labelled "ref.=first", we used the first frame as the p-KLT tracker's reference image and centred its reference patches over the positions of the landmarks in that frame, and in the figures labelled "ref.=previous", we used the previous frame as the reference image, and centred its reference patches over its estimates of the positions of the patches in that frame.	227
B.1	The vertices of the red unit square are left-multiplied by a matrix \mathbf{A} defined as in the text, transforming the square into the blue parallelogram, which has area $ \mathbf{A} $.	239

List of Algorithms

5.1	Constructing the deformation mode model sample data set G_S	127
-	Function $\text{PropagateParticles}(\Theta_{t-1}^{[1:n]}, w_{t-1}^{[1:n]})$ <i>Given the time $t - 1$ particle set, this function samples the time t particle states. .</i>	248
-	Function $\text{CalcGroups}(\Theta_{t-1}^{[1:n]}, w_{t-1}^{[1:n]})$ <i>This function calculates particle hash keys and groups them together.</i>	249
-	Function $\text{CalcDispDistr}(G, \hat{\Psi})$ <i>This function combines discrete, bounded-displacement energy functions to construct a displacement sampling distribution.</i>	250
-	Function $\text{InitDistr}(X)$ <i>This function initialises and returns a structure of temporary variables used in the calculation of the sampling distributions.</i>	250
-	Function $\text{CalcRotDistr}(G, \hat{\Psi}, \Theta)$ <i>This function combines discrete, bounded-displacement energy functions to construct a rotation sampling distribution.</i>	251
-	Function $\text{CalcSkewDistr}(G, \hat{\Psi}, \Theta)$ <i>This function combines discrete, bounded-displacement energy functions to construct a skew sampling distribution.</i>	251
-	Function $\text{CalcScaleDistr}(G, \hat{\Psi}, \Theta)$ <i>This function combines discrete, bounded-displacement energy functions to construct a scale sampling distribution.</i>	251

-	Function <code>UpdateEnergy(ψ, x, Y, l)</code>	
	<i>This function updates the accumulated energies for a sampling distribution.</i>	252
-	Function <code>ConvEnergiesToProbs(Y, X)</code>	
	<i>This function converts the accumulated energies into a sampling distribution.</i>	253
-	Function <code>CalcDeltaEMD($\mathbf{x}_s, \mathbf{x}_t, N, p$)</code>	
	<i>This function calculates the EMD between the delta function and H_p.</i>	254
-	Function <code>CalcUnnormHist($\mathbf{x}_s, \mathbf{x}_t, N$)</code>	
	<i>This function calculates a weighted unnormalised histogram over the squared Euclidean differences between image pixels.</i>	255

Chapter 1

Introduction

1.1 Motivation and Objectives

Coronary heart disease is one of the leading causes of death in men and women. It is caused by atherosclerosis, a condition in which the coronary arteries, which supply blood to the myocardium, become blocked by fatty materials and plaque. According to statistics from the British Heart Foundation, the disease accounted for 22% of male deaths in the UK in 2002 and 17% of female deaths in the same year [101]. Statistics from the American Heart Association [65] indicate that about 1/6th of all deaths in the USA in 2006 were due to this disease, making it the country's leading cause of death.

The two most common procedures used to treat the condition are percutaneous transluminal coronary angioplasties (PTCA) and coronary artery bypass grafts (CABG). The former involves inserting a balloon catheter into the blocked artery and inflating it so as to open the blockage. The latter procedure, used in more serious cases, involves performing a median sternotomy (cracking open the patient's sternum), arresting the heart, performing a cardiopulmonary bypass (CPB) to maintain the flow of blood, harvesting an inessential artery such as the left interior thoracic artery (LITA) and grafting this artery onto the blocked coronary artery so as to bypass the blockage (the grafting process is commonly referred to as an *anastomosis*).

Although CABGs have been in use for over four decades and have had a high success rate, they pose a number of problems for the patient. Firstly, the sternotomy is a traumatic procedure, which results in a long recovery period of up to three months. In addition, a number of serious postoperative complications are associated with CPB, such as organ dysfunction and oedema (swelling) of the myocardium and other tissues due to emboli (vessel blockages) [70, 68]. In an effort to avoid these problems, a number of minimally-invasive alternatives have been developed that can be performed on the beating heart.

The minimally-invasive direct coronary artery bypass (MIDCAB) [11, 110] operation is one such alternative. It is performed through a small incision in the chest (typically 10-12 cm long), alleviating the need for a sternotomy and hence reducing: intraoperative blood loss, the postoperative admission period and the overall recovery period. A mechanical stabiliser is used to reduce motion at the anastomosis site so that the operation can be performed on the beating heart, rendering a CPB unnecessary. MIDCAB is not without its problems however; harvesting the LITA is technically challenging for surgeons due to the limited work space and the restricted field of view. However thanks to recent advances in technology, these limitations can be overcome with the assistance of surgical robots. Such robots allow totally endoscopic coronary artery bypass procedures (TECAB) to be performed, in which the surgeon can view the operating field in 3D via stereo endoscopic cameras. The surgeon is able to manipulate the robot's tools with four to six degrees of freedom through an intuitive telemanipulation interface that eliminates hand tremor and allows for very precise surgical movements. However, TECAB procedures still suffer from a relatively high rate of conversion to either MIDCAB or CABG [32, 56, 27]. Reasons for this include misidentification of the coronary artery and difficulty in locating it due to it being obscured by excessive amounts of fat.

Augmented reality (AR) guidance has the potential to reduce the conversion rate by the superimposition of a preoperatively constructed, accurately registered model of the patient's beating heart onto the surgeon's view of the operating field. The feasibility of such guidance has been demonstrated by previous researchers on phantom and animal data [1, 106, 33], and [63] demonstrated the use of intraoperative ultrasound images for guidance in robot-assisted liver surgery.

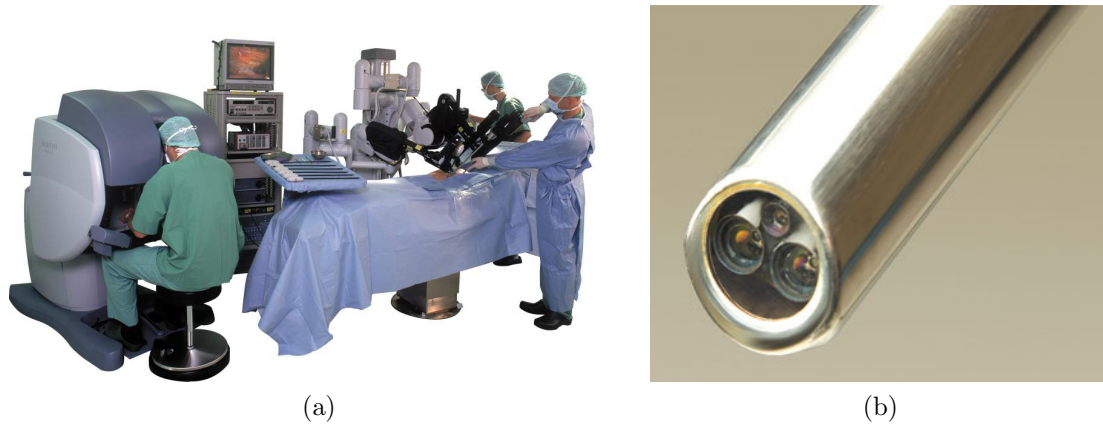


Figure 1.1: The da Vinci system (a) and a close-up of its endoscope (b). The two large circles at the bottom of the endoscope are the stereo cameras, and the smaller circle above them is a light.

The copyright of these images is owned by Intuitive Surgical [51].

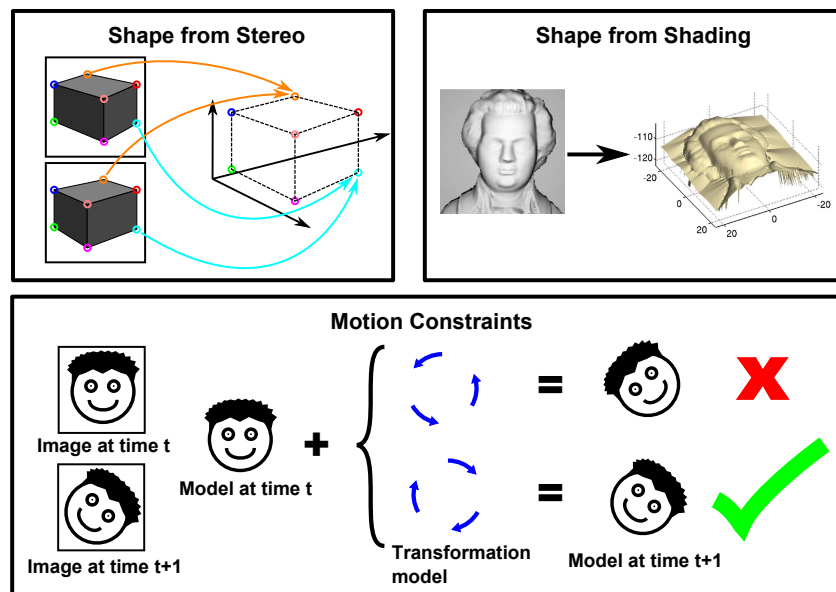


Figure 1.2: This figure shows some of the ways in which images from monoscopic and/or stereoscopic cameras can be processed in order to extract shape information for augmented reality. Shape-from-stereo methods involve exploiting the disparities between corresponding points in a set of simultaneously captured images in order to estimate depth. Shape-from-shading methods use lighting models to estimate depth from changes in surface intensity. Motion information can be used to estimate shape directly, or to place temporal consistency constraints on methods that normally generate independent shape estimates from one time step to the next.

The top-right image is taken from [80].

The work we are presenting in this thesis was carried out as part of a larger project that had the goal of producing such an AR system for the da VinciTM robot, by Intuitive Surgical [51] (shown in figure 1.1). Image guidance in this setting poses particularly severe challenges, as the large homogeneous regions on the surface of the heart are a source of great ambiguity. In addition to this, the myocardium undergoes non-trivial deformations, and the combination of the heart’s moist, shiny appearance and the bright light from the endoscope causes large, distracting specular reflections that must be taken into account. So to achieve accurate and robust guidance results in this setting, it is vital to exploit as much of the available information as possible in order to constrain the process of registering the model to the endoscopic images as tightly as possible.

There are many different types of visual cue that can be used to extract information from endoscopic images, some of which are illustrated in figure 1.2. Accurate estimates of the motion of points on the myocardial surface have many important uses, such as placing temporal consistency constraints on estimates of shape derived from other methods, or even directly estimating shape using structure-from-motion methods ([105, 113]). So for this thesis, we have chosen to focus on the problem of

estimating the trajectories of patches on the surface of the myocardium and the sequences of deformations that they undergo, as observed through a single camera.

Our long-term goal is to have dense motion sequence estimates for all visible points, or better yet, dense 3D motion sequence estimates computed from stereo information. This is a particularly challenging task for myocardial image sequences however, due to the large homogeneous regions, the large amount of deformation that can occur, and, in the case of stereo estimates, the narrow camera baseline. So our plan is to begin by developing methods for estimating the motion of salient patches as accurately and robustly as possible, so that this information can be used to constrain a dense motion estimator in future.

It goes without saying that the use of any motion estimation algorithm for image guidance would require the algorithm to run in real-time. However, due to the great difficulty of the

task, we have chosen not to concentrate on the real-time requirement for the time being, so that we may focus on the equally important issue of how we might go about designing a motion tracker that can be relied upon.

1.2 Contributions

The following is a summary of our main contributions and of the structure of the main thesis chapters:

- **Chapter 3:** In this chapter, we describe a particle-filtering-based framework for tracking patches on the myocardial surface. Under this framework, we present an importance sampling algorithm that is defined in terms of the motion likelihoods of small subregions within the patches. We use a Product-of-Experts-inspired framework to multiplicatively combine this likelihood information into distributions from which we can draw multiple hypotheses of each patch's affine transformations. Furthermore, we provide empirical evidence that ratios of importance sampling probabilities grow exponentially with the number of subregions, and we use this observation to devise a simple method for handling missing likelihood information.
- **Chapter 4:** In this chapter, we describe the methods that we use to calculate likelihoods. We propose a dissimilarity function based on the Earth Mover's Distance, that achieves insensitivity to outliers by ignoring pixel pair differences that are greater than a pre-specified percentile of a weighted pixel pair difference distribution. We also propose a simple method for removing changes in appearance caused by illumination changes.
- **Chapter 5:** This chapter focuses on methods that can be used to learn non-affine myocardial deformation models from a set of training data. The main contribution in this chapter is a B-spline based method for modelling sequences of myocardial deformations, which involves nesting a PCA model of the distribution of static deformed states within a PCA model of the distribution of the B-spline control points.

- **Chapter 6:** This chapter describes our initial attempts to detect and deal with particles that end up in low-probability states, in which they contribute little to the particle filter’s estimate of the patch’s posterior distribution. Our contributions are: the use of “foreground” and “background” likelihood functions to infer when particles are in these low-probability states; a method for estimating the optimal value of a threshold that we use in this inference; the use of our deformation models from chapter 5 to simulate data sets that we can use to estimate the parameters of these likelihood functions and the parameters of the likelihood functions from chapter 4; and a discussion of various methods that could be used to restore particles that are thought to be in low-probability states.

These chapters of original contributions begin after the next chapter, in which we describe some publications by other researchers that are related to our work.

1.3 Statement of Originality

I hereby declare all work presented in this thesis to be my own original research, apart from work that is credited to other researchers.

Chapter 2

Related Work

The work presented in this thesis draws on results from a diverse range of fields, such as function approximation, optimisation, image processing, principal component/functional data analysis, Monte Carlo simulation, statistical inference, and various other areas of statistical analysis. Rather than attempt to condense the relevant background material from all of these fields into a single, inevitably incoherent, chapter, we will reserve this chapter for a discussion of the background material that is most pertinent to our overall aim of estimating myocardial motion, and discuss all other relevant background material during the course of the remaining chapters, as the need arises.

Significant progress in motion estimation and the identification of correspondences between images has been made in restricted environments in which the camera(s) is (are) presumed to be observing a static scene, such as “MonoSLAM” (single-camera *Simultaneous Localisation and Mapping*) [25], and multiview stereo reconstruction (see [96] for a summary of recently proposed methods and an evaluation of their accuracy). Much research has also gone into the design of dense optical flow algorithms (see [6] and [5] for reviews and evaluations), which may make weaker assumptions about the motion of objects in the images so that non-rigid deformations can be handled well, but typically only attempt to model the motion between pairs of images rather than over a whole image sequence. Between these disparate streams of research however, there lies a gulf of problems that require accurate estimates of the long-term

behaviour of objects undergoing complex deformations and occlusions. The application for which we have developed the work presented in this thesis falls into this class.

Sand and Teller presented a method called “Particle Video” in [91] that attempts to find the middle ground between feature tracking algorithms, which typically calculate the trajectories of a sparse set of points independently, and optical flow algorithms. The output of their algorithm is an estimate of the trajectories of a dense set of “particles” (2D points) that are non-uniformly distributed in such a way that the density of particles in any given image region increases with the region’s visual complexity. Given a sequence of images, the algorithm uses a black box optical flow algorithm to calculate initial estimates of the motions of the particles in-between consecutive frames, and refines these estimates by using a variational optimisation algorithm to minimise the sum of a data term, that represents the amount by which a particle’s appearance changes over time, and a distortion term, that represents the dissimilarity between the motion of a particle and the motions of other nearby particles.

Other researchers have proposed methods specifically for the task of myocardial motion estimation. One of the simplest methods is the block-matching method used by Ortmaier et al. in [77] to estimate the displacement of small myocardial regions based on the RMS intensity error.

Stoyanov et al. proposed a more involved method for estimating the 3D translation of myocardial regions viewed through a stereo endoscope in [102], by extending the well-known Lucas-Kanade registration algorithm [66] to take the epipolar constraint into account*. Their method is based on templates defined around stereo feature pairs detected in the initial image pair using the Shi-Tomasi and Maximally-Stable Extremal Region feature detectors [99, 67]. These pairs are triangulated, and their 3D positions in subsequent images are estimated by minimising Lucas-Kanade-style objective functions, defined in terms of the sum of squared differences between the image pairs and linear transformations of the templates that preserve the epipolar constraint.

*The epipolar constraint states that given any stereo pair of images, the set of 3D points that project onto a single point in one image will project onto a line in the other.

The method described by Mountney et. al in [69] begins similarly, by detecting Shi-Tomasi and Difference-of-Gaussian features and tracking them using the Lucas-Kanade method (in 2D). They then take the first n frames (for some adaptively selected n) of the estimated feature trajectories, and use them to construct ID3 decision trees [84] that classify input patches as matches or mismatches. These trees are then used to efficiently track each feature by classifying all possible features within a neighbourhood of its previous position. The published results suggest that the method is more robust than the Lucas-Kanade tracker that provided the training data.

Richa et al. [87] proposed a method for estimating 3D non-rigid myocardial deformation from stereo pairs by calculating parametric mappings on \mathbb{R}^3 which, when projected onto the 2D camera planes, best describe the correspondences between a reference image and the subsequent stereo image pairs. The mappings they used were parameterised as Thin-Plate Splines (which we will discuss in more detail in chapter 5), and again, the objective function that they minimised was defined in terms of the sum of squared differences between the reference and the subsequent image pairs.

Despite the encouraging successful results described by these authors, their methods and experiments have two shortcomings in common. Firstly, all of their published experiments that used real intraoperative images (as opposed to synthetic images, or images of phantom hearts) were performed on mechanically stabilised hearts, i.e. hearts that have had a metal device placed on them that reduces the motion of the anastomosis site (examples of such stabilisers are shown in figures 4.6c and 4.6d on page 104). In practice however, image guidance would be most useful to the surgeon *before* the stabiliser is attached, as the anastomosis site cannot be stabilised before it has been located (see the descriptions of TECAB on the beating heart given in [2, 118]). The myocardium undergoes much more deformation in the absence of a stabiliser, and so it is questionable how well the methods above would perform in this setting.

The second shortcoming (which also applies to the Particle Video method) is that none of these methods attempt to model the uncertainty in their motion estimates. It is especially important to do this when estimating motion prior to stabilisation, when a tracker is most likely to fail

due to the large motions and be in need of information about other likely states. We believe that a statistical approach to tracking, in which some representation of a distribution over the possible feature trajectories is propagated through time, would provide the most natural and mathematically-rigorous mechanisms for modelling the uncertainty.

In our application, as in many practical applications, the dynamics of the features we want to track and the observable image information are highly non-Gaussian and non-linear, and so a statistical algorithm that relies on such restrictive assumptions, such as the classic Kalman filter [55], would almost certainly fare poorly. So instead, we have chosen the more flexible particle filtering algorithm as a discrete framework within which to model these distributions.

Particle filtering is based on the idea of propagating a finite set of weighted hypotheses (“particles”) through time, in order to approximate the posterior distribution of some time-varying random variable of interest, i.e. with the aim of approximating the distribution of that variable at each point in time given all of the observations that were available up till that time point. We will delve into the mathematical details of the algorithm and our implementation of it in the next chapter, and the discussion of different facets of it will continue over most of the rest of the thesis.

Chapter 3

A Product-of-Experts Approach to Importance Sampling

3.1 Synopsis

In this chapter, we shall lay the groundwork for the work that we will present in the remaining chapters of this thesis. We will begin by describing the mathematical assumptions of the particle filtering algorithm that we use to track patches. After this, we shall introduce our first original contribution – an importance sampler from which the particle filter may draw hypotheses about the current state of the patch that we wish to track.

We model the importance sampling distribution as a factor of the posterior distribution’s likelihood term, to avoid the instabilities caused by particles that are occasionally drawn with extremely low importance sampling probability ending up with extremely high weights. We strive to reduce the importance sampling distribution’s uncertainty by multiplicatively combining local likelihood information from different subregions of each particle’s predecessor. We then discuss a generalisation of our importance sampling approach to the case in which some of the local likelihood values are undefined (e.g. as a result of occlusions caused by specular highlights). This generalisation relies on the assumption that ratios of importance sampling probabilities grow exponentially with the number of local likelihood terms that define each im-

portance sampling value. To justify this assumption, we conclude the chapter with an empirical study of the extent to which it seems to hold in the most important case, in which one of the importance probabilities in each ratio is the maximum of the importance sampling distribution that it comes from.

3.2 Affine Patch Tracking with a Particle Filter

The general problem particle filters attempt to solve is that of estimating the posterior distribution of a sequence of hidden states $\Theta_{0:t} = (\Theta_0, \dots, \Theta_t)$, given a sequence of observations $Z_{0:t} = (Z_0, \dots, Z_t)$ that depend on the hidden states. We are interested in tracking a quadrilateral patch through an image sequence, which we model as the problem of estimating the sequence of transformation parameters that determine its trajectory, so that the unknown transformation parameters at time t form the hidden state Θ_t , and the image information at time t forms the observation Z_t .

To improve the well-posedness of the tracking problem and make the necessary calculations tractable, we assume that the patch's transformations are approximately affine. We define the patch to have a default state in which its four vertices $\hat{\mathbf{V}} = (\mathbf{v}_1 \mid \dots \mid \mathbf{v}_4)$ are centred at the origin. $\mathbf{V}(\theta_t)$, the state of the vertices when transformed by θ_t , is then given by

$$\mathbf{V}(\theta_t) = M(\hat{\mathbf{V}}; \theta_t) \triangleq \mathbf{A}(\theta_t)\hat{\mathbf{V}} + (\mathbf{d}(\theta_t) \mid \dots \mid \mathbf{d}(\theta_t)) \ , \quad (3.1)$$

where

$$\begin{aligned} \mathbf{A}(\theta) &\triangleq \mathbf{R}(\theta)\mathbf{S}(\theta)\mathbf{K}(\theta) \\ &= \begin{pmatrix} \cos(\theta_{[\phi]}) & -\sin(\theta_{[\phi]}) \\ \sin(\theta_{[\phi]}) & \cos(\theta_{[\phi]}) \end{pmatrix} \begin{pmatrix} \theta_{[sx]} & 0 \\ 0 & \theta_{[sy]} \end{pmatrix} \begin{pmatrix} 1 & \theta_{[\kappa_x]} \\ \theta_{[\kappa_y]} & 1 \end{pmatrix} \ , \\ \mathbf{d}(\theta) &\triangleq (\theta_{[dx]}, \theta_{[dy]})^T \ , \\ \theta &= [\theta_{[\phi]} \in \mathbb{R}; \theta_{[sx, sy]} \in (0, \infty)^2; \theta_{[\kappa_x, \kappa_y]} \in \{(x, y)^T \in \mathbb{R}^2 : xy < 1\}; \theta_{[dx, dy]} \in \mathbb{R}^2] \ , \end{aligned} \quad (3.2)$$

i.e. $\mathbf{R}(\theta)$, $\mathbf{S}(\theta)$ and $\mathbf{K}(\theta)$ are the rotation, scaling and skewing matrices defined by θ , and $\mathbf{d}(\theta)$ is θ 's translation vector. The matrices have strictly-positive determinants, disallowing reflection and zero scaling.

This representation is non-unique due to the possibility of encoding rotation and scaling information in the skewing parameters $\theta_{[\kappa_x]}$ and $\theta_{[\kappa_y]}$, as shown in figure 3.1, in which the red square is the result of transforming the blue axis-aligned square by

$$\mathbf{A} = \begin{pmatrix} 1 & -0.5 \\ 0.5 & 1 \end{pmatrix},$$

which scales it vertically and horizontally by $\sqrt{\frac{5}{4}}$ and then rotates it clockwise by $\arccos\left(\sqrt{\frac{4}{5}}\right)$.

This can either be achieved with

$$\theta_{[\kappa_y]} = -\theta_{[\kappa_x]} = 0.5, \quad \theta_{[s_x]} = \theta_{[s_y]} = 1, \quad \theta_{[\phi]} = 0,$$

or with

$$\theta_{[\kappa_y]} = \theta_{[\kappa_x]} = 0, \quad \theta_{[s_x]} = \theta_{[s_y]} = \sqrt{\frac{5}{4}}, \quad \theta_{[\phi]} = \arccos\left(\sqrt{\frac{4}{5}}\right).$$

So in practice we store \mathbf{A} rather than $\theta_{[\phi, s_x, s_y, \kappa_x, \kappa_y]}$, and we use a slight variant of the QR decomposition to uniquely calculate the decomposition in eq. (3.2) when we need to use θ 's components. The details of the decomposition are given in appendix B.1. Its uniqueness comes about by requiring $\theta_{[\kappa_y]} = 0$. This component is still useful however, as will be seen in section 3.3.3.

3.2.1 Probabilistic Approximations

We assume that the initial state Θ_0 of the patch is known to be θ_0 (we specify it manually, but it could be chosen by an automatic feature detector). By Bayes' theorem, the posterior

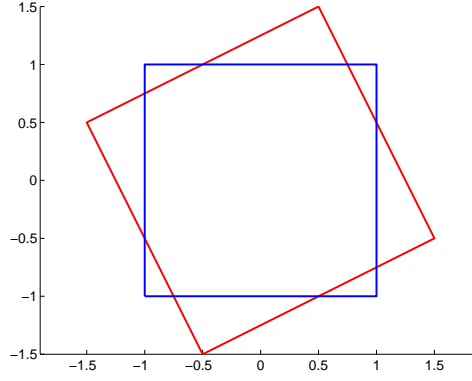


Figure 3.1: An example of the non-uniqueness of the decomposition in eq. (3.2), in which the red square's state can be expressed as a transformation of the blue square's with two different instantiations of θ .

distribution, $\pi_t(\Theta_{1:t})$, can be expressed as follows:

$$\begin{aligned}
 \pi_t(\Theta_{1:t}) &\triangleq P(\Theta_{1:t} \mid Z_{0:t}, \Theta_0) \\
 &\equiv \frac{1}{K_t} P(Z_{1:t} \mid \Theta_{0:t}, Z_0) P(\Theta_{1:t} \mid Z_0, \Theta_0) \\
 &\equiv \frac{1}{K_t} \prod_{i=1}^t P(Z_i \mid Z_{0:i-1}, \Theta_{0:t}) P(\Theta_i \mid \Theta_{0:i-1}, Z_0) , \tag{3.3}
 \end{aligned}$$

where $K_t = P(Z_{1:t} \mid Z_0, \Theta_0)$ is a normalising constant (we write Θ_0 as shorthand for $\Theta_0 = \theta_0$).

To simplify matters, the observation Z_i is often assumed to be conditionally independent of observations $Z_{0:i-1}$ given the state Θ_i , reducing the likelihood terms $P(Z_i \mid Z_{0:i-1}, \Theta_{0:t})$ to $P(Z_i \mid \Theta_i)$. We make the following more general assumption however:

$$\Lambda_i(\Theta_i) \triangleq P(Z_i \mid Z_{0:i-1}, \Theta_{0:t}) = P(Z_i \mid Z_{0:i-1}, \Theta_{0:i}) . \tag{3.4}$$

Retaining $Z_{0:i-1}$ and $\Theta_{0:i-1}$ here does not mean that Z_i is *always* assumed to depend on them. Rather, it should be interpreted as meaning that we *reserve the right* to use some subset of these variables when evaluating Λ_i , if we wish. The reason for this will be made clear in the next section.

Removing Z_i 's conditional dependency on $\Theta_{i+1:t}$ lends eq. (3.3) a Markov-chain-like structure in which the hidden states can be recursively sampled. I.e. it allows us to expand $\Theta_{1:t} = \theta_{1:t}$

drawn from the time t posterior π_t to a sample from π_{t+1} by sampling Θ_{t+1} with probability

$$\frac{K_t}{K_{t+1}} P(Z_{t+1} \mid Z_{0:t}, \Theta_{t+1}, \Theta_{0:t} = \theta_{0:t}) P(\Theta_{t+1} \mid \Theta_{0:t} = \theta_{0:t}, Z_0) . \quad (3.5)$$

The state transition priors $p_i(\Theta_i) \triangleq P(\Theta_i \mid \Theta_{0:i-1}, Z_0)$ are often assumed to follow first-order Markovian dynamics. However we are only interested in the effects of Λ_i and the importance sampling distributions (which will be defined shortly), since this thesis focuses purely on the design of these components. So we assume p_i to be uniform, so that it will contribute no information about Θ_i 's state. Nonetheless, we will retain p_i in our equations for completeness.

Given these assumptions, we are typically interested in estimating the posterior expectations of functions $f(\Theta_{1:t})$, so that we can, for example, calculate moments of $\Theta_{1:t}$:

$$E[f(\Theta_{1:t})] \equiv \frac{1}{K_t} \int f(\theta_{1:t}) \prod_{i=1}^t \Lambda_i(\theta_i) p_i(\theta_i) d\theta_{1:t} , \quad (3.6)$$

or more generally, we may wish to find a Fréchet mean \bar{f}_t of $f(\Theta_{1:t})$

$$\bar{f}_t \equiv \arg \min_{f'} \int d^2(f', f(\theta_{1:t})) \prod_{i=1}^t \Lambda_i(\theta_i) p_i(\theta_i) d\theta_{1:t} , \quad (3.7)$$

where $d(\cdot, \cdot)$ is a metric over f , and in both cases we integrate over each parameter of θ_* over each time step. In practice, such integrals are intractable to calculate, and the normalising constant K_t is generally unknown. Also, sampling directly from the distribution in eq. (3.5) is difficult in general, so the integrals cannot be estimated by Monte Carlo integration.

The standard solution to these problems is to approximate the posterior π_t with a set of n positively-weighted samples (or *particles*), $(\Theta_{0:t}^{[1]}, w_t^{[1]}), \dots, (\Theta_{0:t}^{[n]}, w_t^{[n]})$, each drawn from distributions $I_t^{[j]}(\theta)$, known as the *importance sampling* distributions. The particles are recursively

updated from one time step to the next as follows:

$$\begin{aligned}
(\Theta_0^{[j]}, w_0^{[j]}) &= (\theta_0, n^{-1}) , \\
\Theta_{0:t}^{[j]} &= (\Theta_{0:t-1}^{[j]}, \Theta_t^{[j]} \sim I_t^{[j]}(\theta_t)), \\
w_t^{[j]} &= \frac{w_t'^{[j]}}{\sum_k w_t'^{[k]}} , \\
w_t'^{[j]} &= w_{t-1}^{[j]} \frac{\Lambda_t(\Theta_t^{[j]}) p_t(\Theta_t^{[j]})}{I_t^{[j]}(\Theta_t^{[j]})} ,
\end{aligned} \tag{3.8}$$

where Λ_t may implicitly depend on subsets of the predecessors $\Theta_{0:t-1}^{[j]}$ of $\Theta_t^{[j]}$, as mentioned above. The posterior expectation is then approximated as

$$E[f(\Theta_{1:t})] \approx \bar{f}_t^{[n]} \triangleq \sum_{j=1}^n f(\Theta_{1:t}^{[j]}) w_t^{[j]} , \tag{3.9}$$

and Fréchet means can be estimated by replacing $f(\Theta_{1:t}^{[j]})$ with $d^2(f', f(\Theta_{1:t}^{[j]}))$ and minimising over f' . To weaken the influence of outliers, we only calculate means over the particles that account for the top $y\%$ of the weights (we typically take the top 90%).

In our work, we are interested in calculating weighted means of the affine transformations hypothesised by the particles at each time step. To do this, we calculate the weighted arithmetic mean of the displacements $\mathbf{d}(\Theta_t^{[j]})$, and we estimate Fréchet means $\bar{\mathbf{A}}_t$ of the affine matrices $\mathbf{A}(\Theta_t^{[j]})$ using the method described in [3]:

$$\bar{\mathbf{A}}_t \approx \exp \left\{ \sum_{j=1}^n w_t^{[j]} \log(\mathbf{A}(\Theta_t^{[j]})) \right\} , \tag{3.10}$$

where $\exp(\cdot)$ and $\log(\cdot)$ are the matrix exponential and logarithm respectively, which are both matrix-valued functions of matrices. Efficient and numerically stable algorithms for calculating these functions are given in [40] and [57] (the matrix logarithm algorithm requires the matrix square root, which can be calculated using the method in [9]).

To give a very brief explanation of this result, the matrix logarithm maps matrices to a space that is tangential to the manifold they naturally “live” on (the manifold is a matrix Lie group

and the tangent space is the corresponding Lie algebra). In this space, distances between matrices can be calculated using the Euclidean metric (to a certain extent), and hence matrices can be averaged using the (weighted) arithmetic mean. Finally, the tangent space mean can be mapped back to a matrix in the original space using the matrix exponential. A more thorough and rigorous explanation would lie far outside the scope of this thesis, so the reader is referred to textbooks on differential geometry and Lie algebras for a comprehensive treatment of these matters.

It is worth noting that the matrix logarithm does not always exist, and even when it does, it is not always real. [24] gives a necessary and sufficient condition for any matrix to have a real logarithm, in terms of the parity of the number of Jordan blocks corresponding to each eigenvalue, and [74] specifies cases under which a 2×2 matrix has a real logarithm. We did not encounter any complex matrix logarithms in any of our tests in this thesis, but nevertheless, we describe the geometric significance of these cases in appendix B.2.

With the exception of the departures from the usual treatment of the likelihood function and the prior that we have noted, what we have outlined so far is the simplest common form of particle filtering algorithm, known as the *Sequential Importance Sampling* (SIS) filter [28]. It is known to suffer from degeneracy issues, in which all but one particle eventually end up with 0 weight. Numerous solutions to this have been proposed. The simplest is the *Sampling Importance Resampling* (SIR) algorithm, in which, after calculating the time t particle weights, each particle is resampled, in the sense that it is replaced with particle $\Theta_t^{[j]}$ with probability $w_t^{[j]}$. An often convenient special case of this is the *bootstrap filter* [100, 41], in which the prior is used as the importance sampling distribution.

There are many considerations to be made about the design of the particle filter's constituent distributions before we get as far as dealing with degeneracy issues, so we will not begin to address them until chapter 6. But for more thorough descriptions of some of the methods that have been used to mitigate them, and of other variations of the particle filtering framework, we refer the reader to [28, 4].

3.3 Product-of-Local-Likelihoods Importance Sampling

The question now is: given a set of particles up to time $t - 1$, what distributions should the time t particles be drawn from? A popular choice, used for example in [52], is to model the prior as a low-order vector autoregressive process with Gaussian noise and take it as the importance sampler. In this case, the hidden random variables are treated as vectors $\Theta_t^{[j]}$, and the importance sampler/prior $I_t^{[j]}$ is taken to be the distribution of the n^{th} -order linear process:

$$\Theta_t^{[j]} = \sum_{i=t-n}^{t-1} \mathbf{X}_i \Theta_i^{[j]} + \mathbf{y} + \epsilon, \quad (3.11)$$

where the \mathbf{X}_i are fixed coefficient matrices, \mathbf{y} is a fixed vector and ϵ is a Gaussian random noise vector (see [12] for a detailed analysis of the (zero-mean) first- and second-order case).

One advantage of these models is that sampling from them just involves sampling from a multivariate Gaussian distribution, which can be done very efficiently. Another is that they avoid the problem mentioned in [37] of particles that are occasionally sampled from $I_t^{[j]}$ with very low probability being given extremely large weights by eq. (3.8), leading to a poor estimate of the posterior, which in turn may cause the tracker to drift. This problem does not occur in these models because the importance sampling density is equal to a factor of the posterior, causing it to disappear from the weight update equation:

$$w_t'^{[j]} = w_{t-1}^{[j]} \Lambda_t(\Theta_t^{[j]}) . \quad (3.12)$$

The main disadvantage is that they do not explicitly use the image information from frames 0 to $t - 1$ (although dependencies on it are implicitly created when a particle set is resampled), and worse, they are completely independent of the available time t image information. This means that the Gaussian term needs to have high variance to account for the large errors that will inevitably occur in the deterministic terms when the true motion violates the assumed linearity, which in turn makes the model statistically inefficient and typically requires the use of a large number of particles (e.g. [52] uses 500–1500).

Another problem is that the linear coefficients and Gaussian parameters of the model must either be selected on an ad-hoc basis (e.g. under constant velocity or constant acceleration assumptions) or learnt offline for the specific kind of motion that is to be tracked. The former option is only applicable to very simple tracking problems, whilst the human supervision involved in the latter can be inconvenient, particularly if many different types of motion are to be tracked.

The inspiration for our importance sampling model comes, instead, from the Product-of-Experts model proposed in [49]. The idea behind such models is to multiplicatively combine a set of experts that can each constrain different dimensions of a random variable, so as to reduce the overall model uncertainty.

Figure 3.2a illustrates how combining experts in this way can reduce the kind of uncertainty that leads to the aperture problem*) when estimating the motion of features known to be undergoing the same translation. In this example, the bent purple line at the bottom moves up to the position of the bent line above it, and its motion is to be inferred from likelihood information associated with the regions enclosed by the circles. For each encircled region r on the lower line, suppose we define a binary likelihood function that returns 1 under the hypothesis that r is displaced to an identical circular region, and 0 under all other hypothesised displacements. Then the likelihood of a circle on a straight line segment of the lower shape moving to any point q on a parallel line segment l of the upper shape will be equal for all q on l (apart from q in the regions near l 's endpoints). Thus, the motion of the circles in the directions perpendicular to the line segments they lie on can be determined exactly, but there is high uncertainty in the directions parallel to their line segments (as shown by the bright lines in their likelihood functions). However, enforcing the constraint that all circles undergo the same displacement and combining likelihood values multiplicatively allows the certain component of motion of circles on the rightmost line segment to cancel out the uncertain component of motion of circles on the leftmost line segment, and vice versa. Expressed in terms of the likelihood functions, likelihood values for corresponding displacements will be multiplied,

*)The problem of failing to take sufficient contextual information into account when estimating the motion at a point, leading to high uncertainty along one or more dimensions.

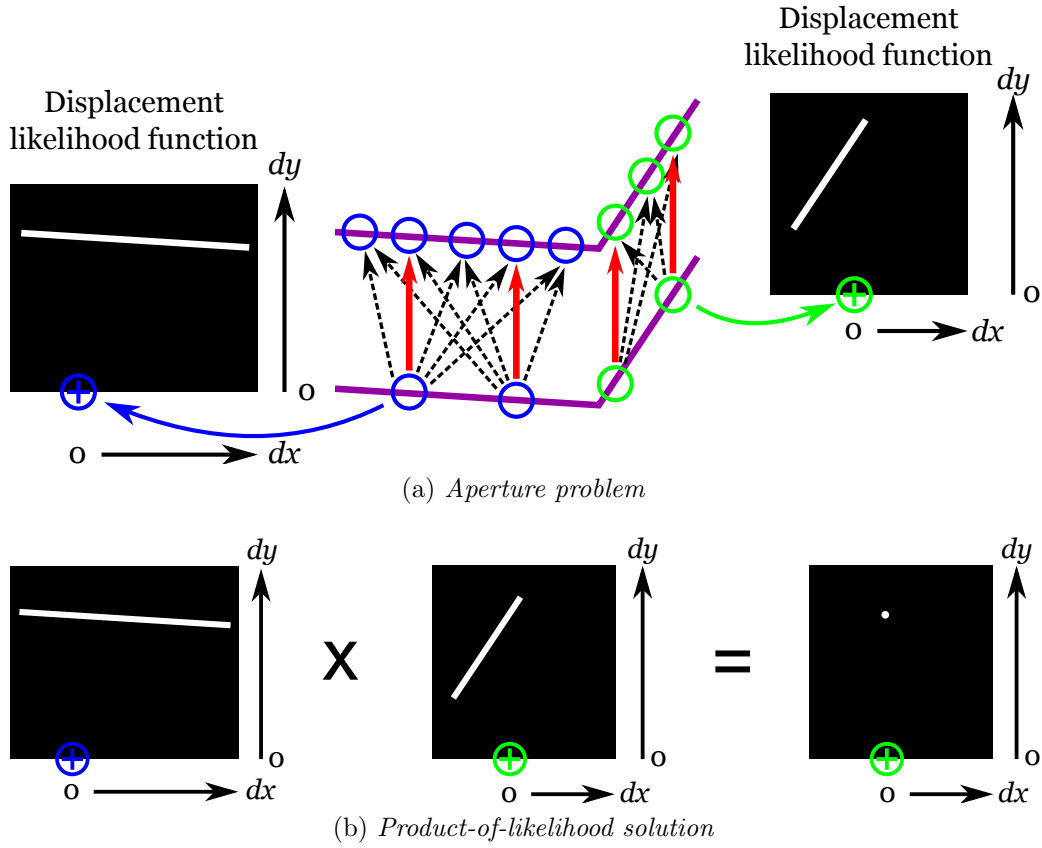


Figure 3.2: (a) The bent purple line at the bottom moves to the bent line at the top. Arrows between encircled regions indicate high-likelihood circle translations; each circle has high uncertainty in the direction parallel to the line segment it lies on. Pairs of circles not joined by arrows have low translation likelihoods. Taking each region’s displacement likelihood as an expert, a Product-of-Experts model gives high probability to the displacement indicated with solid red arrows, and low probability to all other displacements. (b) The pointwise product of a blue circle’s likelihood function and a green circle’s likelihood function annihilates uncertainty, producing a delta function.

producing a delta function that peaks at the intersection of the white lines, as shown in figure 3.2b. Such uncertainty-minimising behaviour can only be achieved by considering information from the current image Z_t , and some subset of the earlier images. This is where our dependency assumption about the likelihood $\Lambda_t(\Theta_t)$ from eq. (3.4) comes into play.

In addition to this behaviour, we would also like our importance sampler to be a factor of the posterior, to avoid the low-importance-sampling-probability problem mentioned above. So, we extract two types of information from $Z_{0:t}$: local information, $Z_{\ell,0:t}$, which, when multiplicatively combined, allows us to draw hypotheses about the patch’s motion from observations about the plausible transformations of its subregions; information about the whole patch, $Z_{L,[0,t]}$, which we use to evaluate the plausibility of the hypothesised patch motions. We then assume that Λ_t

factorises as follows:

$$\Lambda_t(\Theta_t) = P(Z_{\ell,t} \mid Z_{\ell,0:t-1}, \Theta_{0:t}) P(Z_{L,t} \mid Z_{L,0}, \Theta_{[0,t]}) , \quad (3.13)$$

which implies that $Z_{\ell,t}$ is conditionally independent of $Z_{L,t}$. How far from the truth this is depends on the way in which we process Z_t to define $Z_{L,t}$ and $Z_{\ell,0:t}$, and on the joint distribution of the pixels within the patch. A thorough examination of this issue would occupy a whole chapter, and is outside the scope of the issues we are focusing on in this thesis, so we will just proceed under the assumption that they are conditionally independent. The methods we will describe in this chapter treat the manner in which Z_t is processed as a black box, so it can always be changed to better reflect the assumed independence.

Given that the first factor is non-negative, we can take it as an importance sampling distribution over Θ_t if we normalise it with respect to θ_t , i.e. if we define it such that

$$\int P(Z_{\ell,t} \mid Z_{\ell,0:t-1}, \theta_t, \Theta_{0:t-1}) d\theta_t = 1 . \quad (3.14)$$

To make it clear that it is a distribution over Θ_t and that the second factor is a function that evaluates the plausibility of Θ_t , we will use the following synonyms from now on:

$$\begin{aligned} \ell_t(\Theta_t; \Theta_{0:t-1}, Z_{0:t}) &\triangleq P(Z_{\ell,t} \mid Z_{\ell,0:t-1}, \Theta_{0:t}) \\ L_t(\Theta_t; \Theta_0, Z_{[0,t]}) &\triangleq P(Z_{L,t} \mid Z_{L,0}, \Theta_{[0,t]}) . \end{aligned} \quad (3.15)$$

Dependence issues aside, the enforcement of the normality condition in eq. (3.14) is justifiable, since any real function $f(x)$ can be factorised as $f(x) = g(x)h(x)$, where h is an arbitrary real function satisfying $0 < h(x) < \infty$ for all x , and $g(x) = f(x)/h(x)$. In our case, h is ℓ_t , and the lack of formal requirements on Λ_t 's (f 's) form (beyond it being a distribution over Z_t) gives us considerable freedom to define L_t (g) as we please. Extracting a probability distribution factor from Λ_t like this may be compared to the use of sampling distributions in Monte Carlo integration (see [47] for a review).

Taking ℓ_t as our importance sampling distribution for each particle j reduces the unnormalised weights $w_t^{[j]}$ from the expression in eq. (3.8) to

$$w_t^{[j]} = w_{t-1}^{[j]} L_t(\Theta_t; \Theta_0, Z_{[0,t]}) p_t(\Theta_t^{[j]}) . \quad (3.16)$$

The closest work to our importance sampling method (which we will describe in detail in the following subsections) that we are aware of is some articles from the neuroscience literature by Weiss et al. [115, 116] describing Bayesian models of visual motion perception. Their experiments involved calculating MAP (maximum a posteriori) estimates of a 50-parameter deformation field model, and comparing the result to the results perceived by humans. The likelihood term of their model was defined as the product of local likelihood terms from each pixel in a reference image (given the deformation field parameters), and the prior favoured motion that was slow and spatially smooth. The key differences between their approach and ours are that while the deformation fields they estimate can describe more complex motions than the affine transformations we estimate, their model was designed to have a Gaussian posterior (presumably so as to facilitate MAP estimation), which does not allow for the possibility of multiple modes, and their model does not take into account temporal dependencies between the flow field parameters. Furthermore, they only tested their model on the simple synthetic stimuli used in psychophysical visual perception tests, such as rotating barbers' poles, translating plaids, etc., and the moving object in each test image sequence only underwent rigid transformations. So it is questionable how well their method would perform on natural images with complex deformations, like our intraoperative image sequences.

3.3.1 Local Experts

The experts we use to define $\ell_t^{[j]}$ are descriptors of subregions within patch $\mathbf{V}(\Theta_{t-1}^{[j]})$. We define the centres of all subregions within each image to form a regular grid $G = \{i\delta_g : i \in \mathbb{N}\}^2$, where the scalar δ_g is the space between grid vertices, and we are interested in the vertices $G_{t-1}^{[j]} \subset G$ that lie within the boundary of $\mathbf{V}(\Theta_{t-1}^{[j]})$. For each $\mathbf{g} \in G$, the subregion $S(\mathbf{g})$ is defined by the

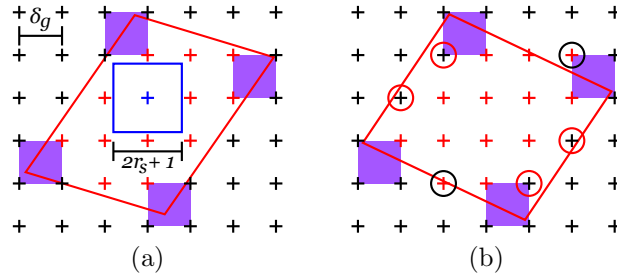


Figure 3.3: (a) The crosses are grid vertices in G . The coloured crosses denote grid vertices lying within the red patch. The blue cross and square denote a subregion of the patch. The patch's vertices lie within the shaded grid squares. (b) After an affine transformation in which the patch's vertices remain within the shaded grid squares, only a few red grid vertices leave the patch (the ones with black circles around them), and only a few black ones enter it (the ones with red circles around them).

image-boundary-aligned square centered at \mathbf{g} with side length $2r_s + 1$ pixels, for some $r_s \in \mathbb{N}$ (see figure 3.3a). We will refer to the i^{th} $\mathbf{g} \in G_{t-1}^{[j]}$ as $\mathbf{g}_{t-1,i}^{[j]}$.

The effect of a hypothesised transformation $\Theta_t^{[j]}$ on any point \mathbf{p} within patch $\mathbf{V}(\Theta_{t-1}^{[j]})$ is a displacement to $\mathcal{M}(\mathbf{p}; \Theta_{t-1}^{[j]}, \Theta_t^{[j]})$, where

$$\begin{aligned} \mathcal{M}(\mathbf{p}; \theta_a, \theta_b) &\triangleq M(M^{-1}(\mathbf{p}; \theta_a); \theta_b) , \\ M^{-1}(\mathbf{B}; \theta_a) &\triangleq \mathbf{A}^{-1}(\theta_a)(\mathbf{B} - (\mathbf{d}(\theta_a) \mid \cdots \mid \mathbf{d}(\theta_a))) . \end{aligned} \quad (3.17)$$

So given a non-negative dissimilarity function $\Delta_{\ell',t}[S, S']$ that compares a descriptor of subregion S (using information from any earlier image between times 0 and $t - 1$) to a descriptor of subregion S' in \mathbf{Z}_t , expert i might give the likelihood of transformation θ_t as

$$\ell'_{t,i}^{[j]}(\theta_t) \equiv \ell'_t(\theta_t; \mathbf{g}_{t-1,i}^{[j]}, \Theta_{t-1}^{[j]}) \triangleq \varphi \left(\Delta_{\ell',t} \left[S(\mathbf{g}_{t-1,i}^{[j]}), \mathcal{M} \left(S(\mathbf{g}_{t-1,i}^{[j]}); \Theta_{t-1}^{[j]}, \theta_t \right) \right]; \alpha_{\ell'}, \gamma_{\ell'} \right) , \quad (3.18)$$

where, with slight abuse of notation, \mathcal{M} is applied to each pixel coordinate in $S(\mathbf{g}_{t-1,i}^{[j]})$, and

$$\begin{aligned} \varphi(\Delta; \alpha, \gamma) &\triangleq e^{-\Psi(\Delta; \alpha, \gamma)} , \\ \Psi(\Delta; \alpha, \gamma) &\triangleq \left(\frac{\Delta}{\gamma} \right)^\alpha . \end{aligned} \quad (3.19)$$

We will give some of the motivating factors behind our choice of a stretched exponential form

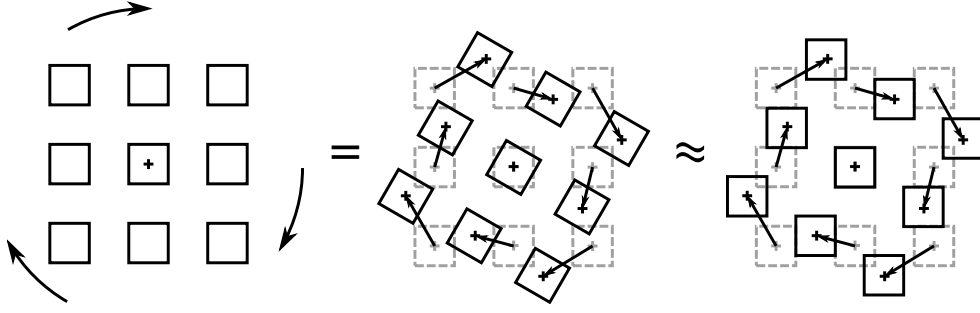


Figure 3.4: A 30° clockwise subregion rotation about the cross in the centre can be approximated by rotating the subregion centres, but preserving the subregions' alignment with the image boundaries.

for φ in section 6.4.3. Viewing φ as an unnormalised distribution over $\Delta_{\ell'}$, the hyperparameters $\alpha_{\ell'}$ and $\gamma_{\ell'}$ control the distribution's kurtosis and inverse decay rate (standard deviation) respectively. In keeping with MRF terminology, we refer to $\Psi(\Delta_{\ell'}; \alpha_{\ell'}, \gamma_{\ell'})$ as φ 's *energy*. More generally, whenever we refer to the energy of any nonnegative function $f(x)$ with supremum $f^\uparrow < \infty$, we mean the nonnegative quantity

$$-\ln \frac{f(x)}{f^\uparrow} . \quad (3.20)$$

We will discuss the precise form of $\Delta_{\ell',t}$ that we used in the next chapter.

3.3.2 Discrete, Bounded-Displacement Energy Functions

For small subregion radii r_s , we can approximate \mathcal{M} 's effects on $S(\mathbf{g}_{t-1,i}^{[j]})$ by uniformly displacing all of its points by

$$\mathbf{d}_{t,i}^{[j]}(\theta_t) \equiv \mathbf{d}(\mathbf{g}_{t-1,i}^{[j]}; \Theta_{t-1}^{[j]}, \theta_t) \triangleq \mathcal{M}(\mathbf{g}_{t-1,i}^{[j]}; \Theta_{t-1}^{[j]}, \theta_t) - \mathbf{g}_{t-1,i}^{[j]} , \quad (3.21)$$

so that $S(\mathbf{g}_{t-1,i}^{[j]})$ maintains its alignment with the image boundaries (e.g. see figure 3.4). This allows us to turn the calculation of $\ell_{t,i}^{[j]}$ on its head by precalculating the energies associated with various displacements of $S(\mathbf{g}_{t-1,i}^{[j]})$ over a discrete, bounded domain, and then referring to these values when evaluating $\ell_{t,i}^{[j]}$.

To make this more precise, let

$$D \triangleq \{i\delta_d : i \in \mathbb{Z}, |i|\delta_d \leq d^\dagger\}^2 \quad (3.22)$$

be the set of subregion displacements under consideration, for some displacement sample spacing factor δ_d . Then we define a displacement energy function $\psi_t^{[j]}(\cdot; \mathbf{g})$ for each $\mathbf{g} \in G_{t-1}^{[j]}$ as

$$\psi_t^{[j]}(\mathbf{d} \in D; \mathbf{g}) \triangleq \Psi(\Delta_{\ell',t}[S(\mathbf{g}), S(\mathbf{g}) + \mathbf{d}]; \alpha_{\ell'}, \gamma_{\ell'}) \quad , \quad (3.23)$$

where, with further abuse of notation, $S(\mathbf{g}) + \mathbf{d}$ displaces every point in $S(\mathbf{g})$ by \mathbf{d} (see figures 3.5a and 3.5b for a real example of $\psi_t^{[j]}$ after negation and exponentiation). We can then approximate $\ell'_{t,i}^{[j]}(\theta_t)$ by interpolating the energies $\psi_t^{[j]}(\cdot; \mathbf{g}_{t-1,i}^{[j]})$ using the closest displacements $\mathbf{d} \in D$ to $\mathbf{d}_{t,i}^{[j]}(\theta_t)$. This of course assumes that $\mathbf{d}_{t,i}^{[j]}(\theta_t)$ lies within D 's boundary. Section 3.3.4 will deal with the case where this is not true.

The advantage of this approximation is that when $\Delta_{\ell',t}$ describes $S(\mathbf{g}_{t-1,i}^{[j]})$'s appearance in terms of \mathbf{Z}_{t-1} 's information only, $\psi_t^{[j]}(\cdot; \mathbf{g}_{t-1,i}^{[j]})$ will be the same for all particles that contain grid point $\mathbf{g}_{t-1,i}^{[j]}$, and so it can be cached and reused. We will discuss the use of earlier images in chapter 6. Note also that while the approximation inaccuracy generally increases with r_s , the accuracy can be improved by calculating $\Delta_{\ell',t}$ with descriptors that are invariant to subregion rotation.

3.3.3 A First-Order Approximation of $\ell_t^{[j]}$

Now that we have discussed how to calculate the local likelihoods $\ell'_{t,i}^{[j]}$, the importance sampling distribution $\ell_t^{[j]}$ can, in theory, be defined as

$$\ell_t^{[j]}(\theta_t; \Theta_{0:t-1}^{[j]}, \mathbf{Z}_{0:t}) = \frac{\prod_{i=1}^{|G_{t-1}^{[j]}|} \ell'_{t,i}^{[j]}(\theta_t)}{\int \prod_{i=1}^{|G_{t-1}^{[j]}|} \ell'_{t,i}^{[j]}(\theta) d\theta} \quad . \quad (3.24)$$

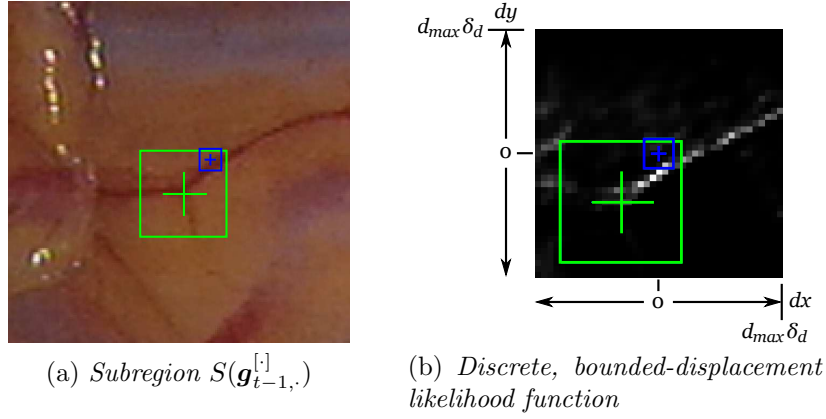


Figure 3.5: (a) A patch on the myocardium (green square), and one of its subregions $S(\mathbf{g}_{t-1}^{[.]})$ (blue square). The time t image is omitted, but it is mostly the result of a small displacement of the shown time $t - 1$ image downwards and to the right. (b) A subregion displacement likelihood function for $S(\mathbf{g}_{t-1}^{[.]})$'s displacements, defined as $e^{-\psi(\cdot; \mathbf{g}_{t-1}^{[.]})}$. Light regions denote high displacement likelihood (low energy), and dark regions denote low likelihood (high energy).

Unfortunately, the computational cost of calculating, or discretising and estimating, the integral would be prohibitively expensive, due to $\Theta_t^{[j]}$'s high dimensionality, and so it would be difficult to sample from $\ell_t^{[j]}$ in this form. This problem can be overcome by Gibbs sampling.

Momentarily considering $\ell_t^{[j]}$ as just a distribution over $\Theta_t^{[j]}$ and omitting its notated dependence on the images for clarity, Bayes' theorem gives us the following factorisation of $\ell_t^{[j]}$ for any permutation $\omega(i \in \{0, \dots, 3\})$ of θ 's four component labels $[\phi]$, $[s_x, s_y]$, $[\kappa_x, \kappa_y]$ and $[d_x, d_y]$:

$$\ell_t^{[j]}(\theta_t) \equiv P\left(\Theta_{t,\omega(0)}^{[j]} = \theta_{t,\omega(0)} \mid \Theta_{t,\omega(1:3)}^{[j]} = \theta_{t,\omega(1:3)}\right) P\left(\Theta_{t,\omega(1:3)}^{[j]} = \theta_{t,\omega(1:3)}\right). \quad (3.25)$$

Hence, if we have a random variable $\Theta_{t,\omega(1:3)}^{[j]} = \theta'_{\omega(1:3)}$ drawn from the second distribution in this factorisation, we only need to sample $\Theta_{t,\omega(0)}$ from the low-dimensional conditional distribution (which is referred to as a *local characteristic* of $\ell_t^{[j]}$ in the Gibbs sampling literature) to create a sample $\Theta_t^{[j]} \sim \ell_t^{[j]}$.

It is easy to approximately sample $\Theta_{t,\omega(0)}^{[j]}$ from the local characteristic in our case. We begin by bounding and discretising $\Theta_{t,\omega(0)}^{[j]}$'s domain, giving a set $X_{\omega(0)}$. Then for each $x \in X_{\omega(0)}$, let $\theta'_{\omega(0)}(x) = x$, and $\theta'_{\omega(1:3)}(x) = \theta'_{\omega(1:3)}$. Vector $\mathbf{d}_{t,i}^{[j]}(\theta'(x))$ is the displacement that $\theta'(x)$ induces on $\mathbf{g}_{t-1,i}^{[.]}$, and the energy $\psi_i(x)$ associated with this displacement can be estimated by

interpolating $\psi_t^{[j]}(\cdot; \mathbf{g}_{t-1,i}^{[j]})$ around $\mathbf{d}_{t,i}^{[j]}(\theta'(x))$, as described in the previous section. So by our Product-of-Experts model, we can approximate the local characteristic $P\left(\Theta_{t,\omega(0)}^{[j]} = x \mid \theta'_{\omega(1:3)}\right)$ by multiplying together the local likelihoods defined by the subregion displacement energies $\psi_i(x)$, i.e.

$$kP\left(\Theta_{t,\omega(0)}^{[j]} = x \mid \theta'_{\omega(1:3)}\right) \approx \tilde{f}(x) = \prod_{i=1}^{|G_{t-1}^{[j]}|} e^{-\min(\psi_i(x), \psi^\uparrow)} = \exp\left\{-\sum_{i=1}^{|G_{t-1}^{[j]}|} \min(\psi_i(x), \psi^\uparrow)\right\}, \quad (3.26)$$

where k is a normalising constant and ψ^\uparrow is an upper bound on the energies that we use to help stave off floating-point underflow. Finally, we evaluate $\tilde{f}(x)$ at all $x \in X_{\omega(0)}$, calculate $k = \sum_x \tilde{f}(x)$, sample a bin x' with probability $k^{-1}\tilde{f}(x')$, and then draw a sample within x' with uniform probability. The accuracy of this approximation increases with the resolution at which X and D are discretised, and it also depends on the error caused by the assumption that subregion points undergo uniform displacement when calculating $\psi_t^{[j]}$.

However the second distribution in eq. (3.25) still has too many dimensions to be sampled from directly. The Gibbs sampling process solves this asymptotically by iteratively sampling from local characteristics like the one above. Given an initial state $\Theta'^{(0)}$, iteration n updates the previous sample as follows:

$$\begin{aligned} \Theta'_{\omega(0)}^{(n)} &\sim P\left(\Theta_{t,\omega(0)}^{[j]} \mid \Theta_{t,\omega(1:3)}^{[j]} = \Theta'_{\omega(1:3)}^{(n-1)}\right), \\ \Theta'_{\omega(1)}^{(n)} &\sim P\left(\Theta_{t,\omega(1)}^{[j]} \mid \Theta'_{\omega(0)}^{(n)}, \Theta'_{\omega([2,3])}^{(n-1)}\right), \\ \Theta'_{\omega(2)}^{(n)} &\sim P\left(\Theta_{t,\omega(2)}^{[j]} \mid \Theta'_{\omega([0,1])}^{(n)}, \Theta'_{\omega(3)}^{(n-1)}\right), \\ \Theta'_{\omega(3)}^{(n)} &\sim P\left(\Theta_{t,\omega(3)}^{[j]} \mid \Theta'_{\omega(0:2)}^{(n)}\right) \end{aligned} \quad (3.27)$$

(we have only fully notated the fixed components of $\Theta_t^{[j]}$ in the first line, to avoid clutter).

Geman and Geman proved a convergence result for Gibbs sampling in [36] which, in our case, implies that $\Theta'^{(n)}$ will converge to a sample from $\ell_t^{[j]}$ as n tends to infinity, regardless of the value of the initial state $\Theta'^{(0)}$ (naturally, the rate of convergence increases if $\Theta'^{(0)}$ is near a

mode of $\ell_t^{[j]}$). The original proof was for discrete random variables, but as explained in [119], it extends to continuous random variables, like $\Theta_t^{[j]}$.

We take the initial state to be $\Theta' = \Theta_{t-1}^{[j]}$ and carry out just a single iteration, under the assumption that $\ell_{t-1}^{[j]}$ is similar to $\ell_t^{[j]}$, so that $\Theta_{t-1}^{[j]}$ should not be too far from a mode of $\ell_t^{[j]}$ (the use of multiple particles helps to mitigate the inaccuracy of this assumption). We start by sampling a displacement vector update \mathbf{d}' over domain D and updating Θ' :

$$\Theta'_{[d_x, d_y]} := \Theta'_{[d_x, d_y]} + \mathbf{d}' + (\mathbf{U} - (0.5, 0.5)^T) \delta_d, \quad (3.28)$$

where \mathbf{U} is a uniform random vector over $[0, 1)^2$ with independent components. Then, using the new value of Θ' to calculate subregion displacements $\mathbf{d}_{t,\cdot}^{[j]}(\Theta')$ in accordance with eqs. (3.1) and (3.2), we sample a rotation angle update ϕ' over domain

$$\hat{R} \triangleq \{i\delta_\phi : i \in \mathbb{Z}, |i|\delta_\phi \leq \phi^\uparrow\}, \quad (3.29)$$

and update Θ' again:

$$\Theta'_{[\phi]} := \Theta'_{[\phi]} + \phi' + (U - 0.5)\delta_\phi, \quad (3.30)$$

where U is a uniform random scalar over $[0, 1)$. We repeat this process to sample a skew update κ' over

$$\hat{K} \triangleq \{(i\delta_{\kappa_x}, 0)^T : i \in \mathbb{Z}, |i|\delta_{\kappa_x} \leq \kappa_x^\uparrow\} \cup \{(0, i\delta_{\kappa_y})^T : i \in \mathbb{Z}, |i|\delta_{\kappa_y} \leq \kappa_y^\uparrow\}, \quad (3.31)$$

and a scale update matrix \mathbf{S}' over

$$\hat{S}(\Theta_{t-1, [s_x, s_y]}^{[j]}) \triangleq \{\text{diag}(\delta_{s_x}^i, \delta_{s_y}^k) : i, k \in \mathbb{Z}, \delta_{s_x}^{|i|} \leq s^\uparrow, \delta_{s_y}^{|k|} \leq s^\uparrow\}, \quad (3.32)$$

which updates Θ' 's scale parameters as:

$$\Theta'_{[s_x, s_y]} := \text{diag}(\delta_{s_x}^{U_0-0.5}, \delta_{s_y}^{U_1-0.5}) \mathbf{S}' \Theta'_{[s_x, s_y]}, \quad (3.33)$$

where, again, U_0 and U_1 are independent uniform random scalars over $[0, 1)$. Finally, the new

hypothesised hidden state is $\Theta_t^{[j]} = \Theta'$.

The parameters $\delta_\phi > 0$, $\delta_{\kappa_\cdot} > 0$ and $\delta_s > 1$ determine the discretisation resolutions of \hat{R} , \hat{K} and \hat{S} respectively, and ϕ^\uparrow , κ_x^\uparrow , κ_y^\uparrow and s^\uparrow determine the boundaries of these sets*. We calculate the discretisation resolution parameters so that an increase in rotation of δ_ϕ , an increase in skew of δ_{κ_\cdot} , or an increase in scale parameter $\Theta_{t-1,[s]}^{[j]}$ by a factor of δ_s causes the furthest vertex from the centre of the patch to move by one pixel. The details of these calculations are given in appendix B.3.

Again, if $\Delta_{\ell',t}$ only uses Z_{t-1} 's information in the description of each $S(\mathbf{g}_{t-1,i}^{[j]})$, we can make an approximation that allows us to cache the conditional distributions for displacement updates and reuse them when processing other particles. The approximation is based on the observation that if you were to move the vertices of the red patch in figure 3.3a within the shaded grid squares that they lie in, a few grid points near the patch's boundary might enter or leave the patch, but most grid points would not cross the boundary, as demonstrated in figure 3.3b. Given this, for any patch defined by a time $t-1$ particle transformation $\Theta_{t-1}^{[j]}$, the four grid squares that the patch's vertices lie in can be taken as a hash table key that roughly identifies grid points $G_{t-1}^{[j]}$. So at the beginning of time step t , we calculate these hash keys and group together all particles sharing a key value, as in figure 3.6. For each group Q with key k_Q , we calculate the weighted average of the affine matrices $\mathbf{A}(\cdot)$ and displacements $\mathbf{d}(\cdot)$ over Q 's particles (using the particle weights), giving an average transformation $\bar{\Theta}_Q$. Then for each particle j in Q , we define $G_{t-1}^{[j]}$ as the grid points within the boundary of $\mathbf{V}(\bar{\Theta}_Q)$. Thus, the conditional distribution for displacement updates over D will be the same for all particles in Q , and can be cached by key k_Q .

3.3.4 Local Likelihood Energy Problems

The energies $\psi_i(x)$ involved in the calculation of each conditional distribution $k^{-1}\tilde{f}(x)$ may be undefined for one of the following reasons:

* κ_x^\uparrow and κ_y^\uparrow may be different, as Θ' must satisfy $\Theta'_{[\kappa_x]}\Theta'_{[\kappa_y]} < 1$, so as to ensure that the determinant of the hypothesised transformation remains strictly-positive.

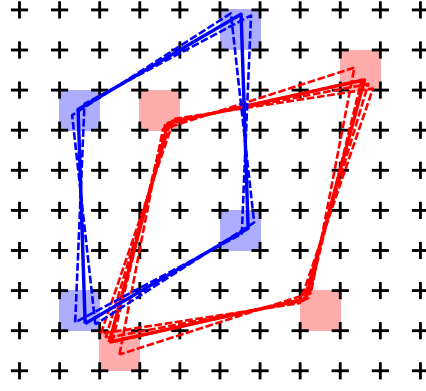


Figure 3.6: Dashed parallelograms denote patches defined by time $t - 1$ particles. Patches sharing a hash key (defined by the shaded squares that patch vertices lie in) are grouped together by colour, and the group averages are represented by the solid parallelograms.

- a) the displacement $\mathbf{d}_{t,i}^{[j]}(\theta'(x))$ may lie outside of D 's boundary;
- b) subregions $S(\mathbf{g}_{t-1,i}^{[j]})$ and/or $S(\mathbf{g}_{t-1,i}^{[j]} + \mathbf{d})$ (for some $\mathbf{d} \in D$) may have high proportions of unobservable pixels, either as a result of a mask indicating occlusion by specular highlights/foreground objects, or as a result of the subregions lying on or outside the image boundary.

Intuitively, a Product-of-Experts model of $\ell_t^{[j]}$ should be able to handle the absence of a few energy values, especially if the defined energy values still provide reliable information (e.g. if the subregions for which energies are defined lie on non-parallel edges, as in figure 3.2a). So we would like a way of handling undefined energies that is equivalent to eq. (3.26) when all the desired information is available, and that degrades gracefully as information is lost.

In the case where m of the energy functions $\psi_i(x)$ are defined everywhere in their domains and the remaining $|G_{t-1}^{[j]}| - m$ energy functions are undefined everywhere in their domains, we can simply omit the undefined terms from eq. (3.26), as this is equivalent to only using m of the subregions around the patch's grid points $G_{t-1}^{[j]}$. As far as that equation is concerned, this case is equivalent to the case in which there are exactly m defined energy terms for each transformation state x in the domain of the conditional distribution $k^{-1}\tilde{f}(x)$, but some energy functions are only partially defined. So we may handle this latter case in the same way, omitting all the undefined terms from eq. (3.26). This point is illustrated in figure 3.7.

degrees of uncertainty, and hence giving our importance sampler its uncertainty-minimising behaviour.

Although this exponential rate of probability ratio growth is a possibility, it is not guaranteed to occur in all cases. So we will now examine the conditions under which exponential ratio growth does occur, and conditions under which we can reasonably assume that it would occur if all of the energy terms were observable. In the course of doing so, we will present a generalisation of eq. (3.26) that reduces to it when all of the energy terms are defined and that, in a sense, preserves the possibility of exponential ratio growth when there are varying numbers of undefined energy terms.

Exponential Probability Ratio Growth

Taking X as the domain of the conditional sampling function $k^{-1}\tilde{f}(x)$ as before, let ρ_i , $1 \leq i \leq |G_{t-1}^{[j]}|$, define an arbitrary ordering over the subregion centres $G_{t-1}^{[j]}$ of a time $t - 1$ patch $\mathbf{V}(\Theta_{t-1}^{[j]})$, and hence over the energies $\psi_i(x)$ associated with any given $x \in X$. The probability of sampling transformation parameter $x \in X$ from $k^{-1}\tilde{f}$ is determined up to \tilde{f} 's normalising factor k^{-1} by the likelihood energies $\psi_1(x), \psi_2(x), \dots$. Given this energy term ordering, we define the n^{th} **partial mean energy** $\bar{\psi}_n(x)$ for a state x as

$$\bar{\psi}_n(x) \triangleq \frac{1}{n} \sum_{i=1}^n \psi_i(x) . \quad (3.34)$$

By eq. (3.26), we can then redefine $\tilde{f}_n(x)$, the unnormalised conditional probability of sampling x when using the first n energies, as follows:

$$\tilde{f}_n(x) \equiv e^{-n\bar{\psi}_n(x)} . \quad (3.35)$$

This suggests a new interpretation of \tilde{f} 's energies as the product of an exponentiation rate, $\bar{\psi}_n$, and an order of magnitude, n .

Generally speaking, it would be preferable to use as many energy terms as possible. So we

can view the normalised sampling distribution under n energy terms as an approximation of a normalised sampling distribution under an infinite number of energy terms, i.e.

$$k_n^{-1} \tilde{f}_n(x) \approx k_\infty^{-1} \tilde{f}_\infty(x) \triangleq \lim_{m \rightarrow \infty} k_m^{-1} \tilde{f}_m(x) , \quad (3.36)$$

where k_n is \tilde{f}_n 's normalising factor. $k_n^{-1} \tilde{f}_n(x)$ converges in n if the partial mean energies all converge, i.e. if

$$\bar{\psi}_\infty(x) \triangleq \lim_{n \rightarrow \infty} \bar{\psi}_n(x) \quad (3.37)$$

exists for all $x \in X^*$. When convergence occurs, $k_\infty^{-1} \tilde{f}_\infty$ is a distribution that gives equal probability $p > 0$ to all states x at which $\bar{\psi}_\infty(x)$ is minimal, and probability 0 to all other states.

Suppose that the infinite sequences of energy terms involved in the definition of $\bar{\psi}_\infty$ correspond to a countably infinite set of subregion centres $G_{t-1}^{[j]}$, and that the ordering imposed on them by ρ_i gradually increases the number of subregion centres that fall within every neighbourhood of every point in the patch. This increase in subregion density increases the density of the image samples used in the definition of $k_n^{-1} \tilde{f}_n(x)$. Since myocardial deformations are generally smooth, it seems reasonable to assume that $\bar{\psi}_n$, and hence $k_n^{-1} \tilde{f}_n$, will converge in n as the density of the image samples increases.

Assuming that $k_n^{-1} \tilde{f}_n$ does converge in n , the ratio of values of $k_n^{-1} \tilde{f}_n$ at states that $k_\infty^{-1} \tilde{f}_\infty$ assigns different probabilities to must eventually tend to $p/0 = \infty$ (or equivalently, to $0/p = 0$). In the following investigation of the conditions under which these ratios diverge at an exponential rate, we will consider \tilde{f}_n 's values at two transformation states $x_a \in X$ and $x_b \in X$, and we will adopt the following shorthands:

$$\begin{aligned} \psi_{a,i} &\triangleq \psi_i(x_a) , & \psi_{b,i} &\triangleq \psi_i(x_b) , \\ \bar{\psi}_{a,n} &\triangleq \bar{\psi}_n(x_a) , & \bar{\psi}_{b,n} &\triangleq \bar{\psi}_n(x_b) . \end{aligned} \quad (3.38)$$

*Note that $\bar{\psi}_\infty(x)$ is always finite when it exists, since the energy terms always lie in $[0, \psi^\dagger]$

Ignoring the ambiguous case in which two parameters should be sampled from $k_n^{-1} \tilde{f}_n$ with equal probability, suppose, without loss of generality, that the inequality

$$\tilde{f}_n(x_a) > \tilde{f}_n(x_b) \quad (3.39)$$

holds for some value of n . Taking negated logs, this then implies that

$$\bar{\psi}_{a,n} < \bar{\psi}_{b,n} \quad (3.40)$$

Furthermore, since

$$\psi_a^\downarrow \leq \bar{\psi}_{a,n} \text{ and } \bar{\psi}_{b,n} \leq \psi_b^\uparrow, \quad (3.41)$$

where

$$\psi_a^\downarrow \triangleq \inf_i \psi_{a,i} \text{ and } \psi_b^\uparrow \triangleq \sup_i \psi_{b,i}, \quad (3.42)$$

an upper bound on the ratio of the probabilities of sampling x_a and x_b is

$$\frac{\tilde{f}_n(x_a)}{\tilde{f}_n(x_b)} \leq \frac{e^{-n\psi_a^\downarrow}}{e^{-n\psi_b^\uparrow}} = e^{n(\psi_b^\uparrow - \psi_a^\downarrow)}, \quad (3.43)$$

which clearly grows exponentially in n . For a lower bound on the ratio to also grow exponentially in n , there must be a real number $\alpha > 0$ and a natural number n' such that

$$\begin{aligned} \forall n > n' \quad & (\bar{\psi}_{b,n} - \bar{\psi}_{a,n} \geq \alpha) \\ \Rightarrow \forall n > n' \quad & (n(\bar{\psi}_{b,n} - \bar{\psi}_{a,n}) \geq n\alpha), \end{aligned} \quad (3.44)$$

where we use n' to allow for fluctuations in the sign of $\bar{\psi}_{b,n} - \bar{\psi}_{a,n}$ when the number of energy terms is small. When this holds, an exponential lower bound is given by

$$\forall n > n' \quad \left(e^{n\alpha} \leq e^{n(\bar{\psi}_{b,n} - \bar{\psi}_{a,n})} = \frac{\tilde{f}_n(x_a)}{\tilde{f}_n(x_b)} \right). \quad (3.45)$$

Considering the fact that we are not always able to observe the same number n of energy terms for each x , the convergence of $\bar{\psi}_n(x)$ in n is a useful assumption to make when trying to evaluate

$\tilde{f}(x)$ as it, together with the assumption that ratios of \tilde{f}_n would grow exponentially in n if we could observe an arbitrary number of energy terms for each x , suggests that we can use the observable energy terms to calculate estimates $\bar{\psi}'(x)$ of $\bar{\psi}_\infty(x)$ for each x , and then combine these estimates with eq. (3.35) to redefine $\tilde{f}(x)$ as

$$\tilde{f}(x) = e^{-\bar{n}\bar{\psi}'(x)} , \quad (3.46)$$

where the order of magnitude \bar{n} is constant over all x and reflects the number of observable energies at each x . An obvious choice for \bar{n} would be the mean number of observable energies. Similarly, if only m_a energy terms were observable for x_a and m_b for x_b , we could use the partial means $\bar{\psi}_{a,m_a}$ and $\bar{\psi}_{b,m_b}$ as estimates for $\bar{\psi}_{a,\infty}$ and $\bar{\psi}_{b,\infty}$.

This definition of $\tilde{f}(x)$ makes use of the exponential ratio growth model in the sense that

$$\frac{\tilde{f}(x_a)}{\tilde{f}(x_b)} = e^{\bar{n}(\bar{\psi}_{b,m_b} - \bar{\psi}_{a,m_a})} \quad (3.47)$$

may grow exponentially with \bar{n} , depending on the behaviour of $\bar{\psi}_{a,m_a}$ and $\bar{\psi}_{b,m_b}$ as they converge in m_a and m_b . In particular, if, as a generalisation of eq. (3.44), $\bar{\psi}_{a,m_a}$ and $\bar{\psi}_{b,m_b}$ are always separated by a margin $\alpha' > 0$ when m_a and m_b are sufficiently large, i.e. if

$$\min(m_a, m_b) > m' \Rightarrow \bar{\psi}_{b,m_b} - \bar{\psi}_{a,m_a} > \alpha' \quad (3.48)$$

for some m' , then $e^{\bar{n}\alpha'}$ is a lower bound on $\tilde{f}(x_a)/\tilde{f}(x_b)$ that grows exponentially in \bar{n} .

The case in which the exponential ratio growth assumption is most helpful in minimising uncertainty is the case in which $\bar{\psi}_{a,m_a} \ll \bar{\psi}_{b,m_b}$, suggesting that x_a is closer to a mode of \tilde{f}_∞ than x_b . In this case, provided that m_a and m_b are large enough for the partial means to be insensitive to fluctuations in their energy term sequences, the energies $\psi_{a,1:m_a}$ are likely to be in close agreement, and the sign of $\bar{\psi}_{b,m_b} - \bar{\psi}_{a,m_a}$ is likely to be the same as the sign of $\bar{\psi}_{b,m'_b} - \bar{\psi}_{a,m'_a}$ for all $m'_a > m_a$ and $m'_b > m_b$. So it seems reasonable to assume that

$$\min(m_a, m_b) > m' \wedge \bar{\psi}_{a,m_a} \ll \bar{\psi}_{b,m_b} \Rightarrow \forall m'_a, m'_b \geq \min(m_a, m_b) \quad (\bar{\psi}_{a,m'_a} \ll \bar{\psi}_{b,m'_b}) , \quad (3.49)$$

and to take the truth of the antecedent (for some appropriately large m') as evidence that $\tilde{f}(x_a)/\tilde{f}(x_b)$ is an exponential function of \bar{n} (since the truth of eq. (3.49) implies the existence of a margin between $\bar{\psi}_{a,m_a}$ and $\bar{\psi}_{b,m_b}$, as in eq. (3.48)).

When $\bar{\psi}_{a,m_a}$ and $\bar{\psi}_{a,m_b}$ are close in value, there is less certainty as to what their ordering will be as $\min(m_a, m_b)$ increases, and so a variant of eq. (3.49) in which \ll is replaced with $<$ is less likely to be true. However, if both x_a and x_b are far from the modes, an exponential growth assumption will still lead to the most important behaviour of reducing the probabilities of x_a and x_b relative to the probabilities near the modes. If they are close to modes, then we have an ambiguous case that we cannot reliably deal with without additional information.

The final case to be dealt with is the pathological case in which no energies are observable for some states $X_u \subseteq X$. In this case, we have no information about the states in X_u , and so the best we can do is to take the most noncommittal approach in which no state of X_u is considered more likely than another, and the probability of sampling from X_u maximises the uncertainty of \tilde{f} . In [53], Jaynes advocates the use of Shannon's information entropy,

$$H \triangleq - \sum_{x \in X} k^{-1} \tilde{f}(x) \ln(k^{-1} \tilde{f}(x)) , \quad (3.50)$$

as the measure of uncertainty to maximise in such situations, and he sketches a derivation of the measure from three weak axioms that an uncertainty measure should satisfy. The most intuitively necessary properties of H for our purpose are that it gives a measure of 0 to delta functions, that it varies continuously with the probabilities of the distribution in question, and that it is maximised by the uniform distribution. Thus, the problem of selecting an uncertainty-maximising probability to assign to the $x \in X_u$ can be thought of as the problem of selecting a probability that will bring \tilde{f} as close as possible to a uniform distribution.

Handling Underflow

Whilst the preceding section focused on the issue of too few defined energy terms contributing to the probability that $k^{-1} \tilde{f}$ assigns to some state $x \in X$, the opposite problem of too many defined

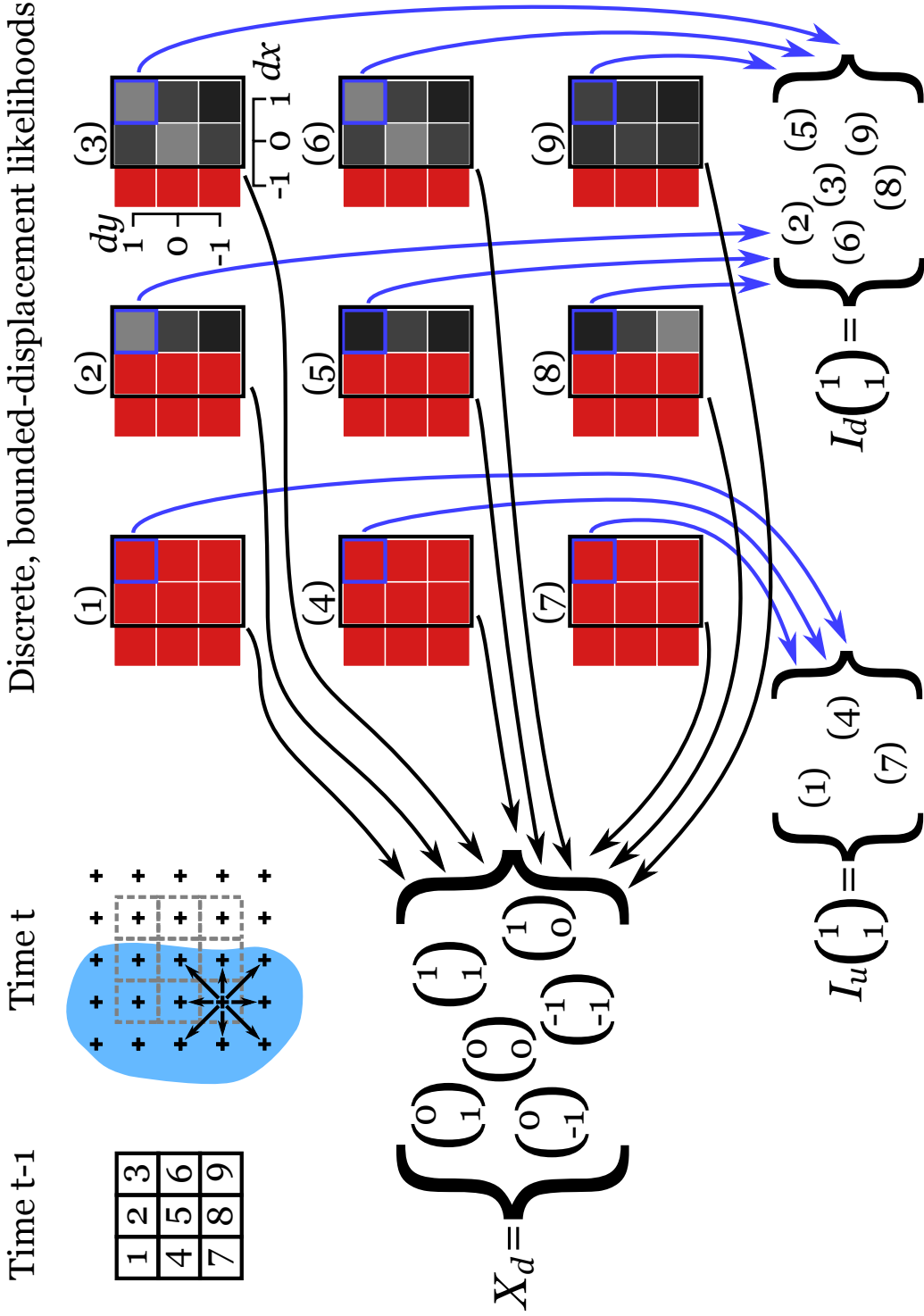


Figure 3.8: The 3×3 numbered time $t-1$ subregions either remain in their current position or move to one of the 8 positions surrounding them, as shown in the time t diagram. The pale blue blob indicates a masked out region of the time t image, which destroys all of the likelihood information for subregions (1), (4) and (7), as indicated by the red squares in the displacement likelihood grids on the right. In these grids, the black loop around the rightmost 6 squares indicates displacements for which at least one likelihood value is defined, and the squares indicated in purple highlight the likelihoods for displacement $(1, 1)^T$.

energy terms contributing to $\tilde{f}(x)$ is equally important, as it can result in underflow (rounding to zero) when the energies are negated and exponentiated. For double precision floating-point numbers, the maximum number of subregions that can contribute maximal energy before the likelihood product falls into the denormalised (i.e. reduced precision) range is $\lfloor \frac{1022 \ln 2}{\psi^\uparrow} \rfloor$, and the maximum number that can contribute maximal energy before underflow occurs is $\lfloor \frac{1074 \ln 2}{\psi^\uparrow} \rfloor$ (this follows from the fact that the smallest normalised number is 2^{-1022} and the smallest denormalised number > 0 is 2^{-1074} – see [39]). We set $\psi^\uparrow = 30$ for all results presented in this thesis, which means that 24 subregions are sufficient to cause denormalisation, and 25 are sufficient to cause underflow. Working in the denormalised range is undesirable, for reasons related to numerical accuracy, but underflow can have the destructive effect of causing states of X that should have high probability after normalising \tilde{f} to end up with 0 probability, and can even cause $k^{-1}\tilde{f}$ to be an invalid distribution (when underflow occurs at all states of X).

The simplest way to prevent \tilde{f} from assigning denormalised values to the highest probability states is to uniformly rescale the values it assigns to all states by adding a value c to all of the negated partial mean energy terms. This is a valid transformation, as any factor by which \tilde{f} is scaled will disappear after normalisation. The precise value chosen for c is not critical, but needless to say, it should not be so high that it causes \tilde{f} to overflow! In our implementation, we define c as the minimum partial mean energy over each state of X , so that \tilde{f} always assigns a value of 1 to the modes before normalisation. Under this choice of c , multiplying \tilde{f} by k^{-1} could only ever cause the probabilities at the modes to enter the denormalised range if $k > 2^{1022}$, which, even if \tilde{f} was uniform, would require $|X|$ to be about 1000 orders of magnitude (in base 2) larger than it typically is!

If \tilde{f} assigns a small value ϵ to a non-modal state, the precise condition under which $k^{-1}\epsilon$ will be a denormalised number depends on the values that \tilde{f} assigns to the other states. However, we can estimate a lower bound ϵ^\downarrow on ϵ , above which multiplying by k^{-1} does not cause denormalisation, by assuming the most extreme case, in which \tilde{f} assigns value 1 to the remaining $|X| - 1$ states,

in which case $k = |X| + \epsilon - 1$. $k^{-1}\epsilon$ is not a denormalised number if

$$\frac{\epsilon}{|X| + \epsilon - 1} \geq 2^{-1022} \quad (3.51)$$

in floating-point arithmetic, suggesting that

$$\epsilon^\downarrow \approx (|X| - 1)2^{-1022} . \quad (3.52)$$

The sampling distributions that we calculate for displacement have the largest domains X . E.g. in the next section's results $|X| = 1681$, for which $\epsilon^\downarrow \approx 2^{-1011}$. Given that our rescaled \tilde{f} always assigns value 1 to the modes, the probability of sampling any state to which \tilde{f} assigns a value less than this bound is so low that it is immaterial whether or not these values of \tilde{f} enter the denormalised range or underflow.

Summary

To summarise all of this mathematically, for each $x \in X$, let $I_d(x)$ contain $G_{t-1}^{[j]}$'s grid point indices $i \in \{1, \dots, |G_{t-1}^{[j]}|\}$ for which $\psi_i(x)$ is defined, and let $I_u(x)$ contain the grid point indices for which $\psi_i(x)$ is undefined. In addition, let $X_d = \{x \in X : |I_d(x)| > 0\}$ (see figure 3.8 for an illustration of the relationship between masked out image regions and the construction of these sets). We redefine $\tilde{f}(x)$ as

$$\tilde{f}(x) = \begin{cases} e^{-\bar{n}(\bar{\psi}(x) - \bar{\psi}^\downarrow)} & , \quad x \in X_d , \\ e^{-\psi_u} & , \quad \text{otherwise} , \end{cases} \quad (3.53)$$

where

$$\begin{aligned} \bar{n} &\triangleq \frac{1}{|X_d|} \sum_{x' \in X_d} |I_d(x')| , & \bar{\psi}(x) &\triangleq \frac{1}{|I_d(x)|} \sum_{i \in I_d(x)} \min(\psi_i(x), \psi^\uparrow) , & \bar{\psi}^\downarrow &\triangleq \min_{x \in X_d} \bar{\psi}(x) , \\ \psi_u &\triangleq \begin{cases} \frac{-\sum_{x' \in X_d} \tilde{f}(x') \ln(\tilde{f}(x'))}{\sum_{x' \in X_d} \tilde{f}(x')} & , \quad |X_d| > 0 , \\ 0 & , \quad \text{otherwise} . \end{cases} \end{aligned} \quad (3.54)$$

If $\psi_i(x)$ is defined for all i and x , $\bar{n} = |G_{t-1}^{[j]}| = |I_d(x)|$ for all x , and $\tilde{f}(x)$ is scaled by $e^{\bar{n}\bar{\psi}^\dagger}$ for all x . So up to a scaling factor, eq. (3.53) reduces to eq. (3.26) as required. In appendix B.4, we prove that when $x \notin X_d$, $\tilde{f}(x) = e^{-\psi_u}$ maximises the entropy of $k^{-1}\tilde{f}$. Note that in general, when defining the order of magnitude term \bar{n} , we only average the number of observable energies over each $x \in X_d$, since we treat $x \in X - X_d$ separately.

The whole process of constructing the importance sampling distributions and using them to propagate the states of the particles is summarised in the function `PropagateParticles` and its helper functions, which are given in appendix C.

3.4 Results: Exponential Likelihood Ratio Growth Study

The approach to handling undefined local likelihood energies that we presented in the previous section relied on a number of assumptions about the behaviour of the partial means as the number of defined energy terms grows. Our aim in this section is to provide empirical evidence for the correctness of these assumptions. In particular, the two most significant assumptions that we wish to validate are:

1. the assumption that the i^{th} partial mean $\bar{\psi}_{a,i}$ of any state x_a is likely to converge as i increases;
2. eq. (3.49), which, if correct, places an exponential lower bound on the growth of the ratio between the importance sampling probabilities of pairs of states x_a and x_b with very different m_a^{th} and m_b^{th} partial mean energies.

To test the validity of the two assumptions, we performed a series of experiments based on the defined parts of the discrete, bounded-displacement energy functions $\psi_t^{[j]}$ that determine the importance sampling distributions for the particles of 4 patches that we tracked in 2 videos, each from a different patient. All of the experiments in this thesis are based on the same two videos, which we refer to as “230108.FO1” and “180608.HS”*. To track the patches, we defined the local likelihood ℓ' and patch likelihood L in terms of the dissimilarity functions Δ'_ℓ and Δ_L that we will discuss in the next chapter, and we determined their hyperparameters ($\alpha_{\ell'}$ and $\gamma_{\ell'}$ for ℓ' , and α_L and γ_L for L) using the simulation-based maximum likelihood approach that we will describe in chapter 6.

In that chapter, we will also describe methods for resampling the particle sets when particles begin to drift. We did not perform any resampling for the tests we are about to describe however, so as to maximise the number of distinct importance sampling distributions that each patch’s particles used (bearing in mind that we cache and reuse the displacement distributions for patches that share hash keys, as we described earlier), and hence maximise the amount

*See the results section at the end of chapter 5 for a statistical summary of the amount of motion/deformation in these videos.

Parameter	Value	
	230108.FO1	180608.HS
No. particles	50	50
Initial patch side length	100p	80p
Grid point spacing, δ_g	15p	10p
Subregion radius, r_s	8p	7p
$\psi_t^{[j]}$ sample spacing, δ_d	2p	2p
Max. displacement, d^\uparrow	40p	40p
Max. rotation, ϕ^\uparrow	10°	10°
Max. scale, s^\uparrow	1.25	1.25
Max. skew, $\kappa_x^\uparrow, \kappa_y^\uparrow$	≤ 0.2	≤ 0.2
Max. energy, ψ^\uparrow	30	30
No. \mathcal{E} histogram bins, N	256	256
\mathcal{E} outlier cutoff for ℓ', p	0.5	0.5
\mathcal{E} outlier cutoff for L, p	0.75	0.75

Table 3.1: The values of the particle filter parameters that we manually defined. The unit p stand for “pixels”. Unless otherwise stated, we use the same values for all tests in this thesis. *Note: \mathcal{E} is a dissimilarity function that we will use to define $\Delta_{\ell'}$ and Δ_L in the next chapter.*

Seq.	Percentile				
	0%	25%	50%	75%	100%
230108.FO1	20	41	46	54	92
180608.HS	13	48	60	69	108

Table 3.2: The percentiles of the number of subregions that defined the particles’ sampling distributions.

of data available for analysis. The fact that not resampling means that some particles would have been in states of low posterior probability does not matter for these tests, nor does the accuracy with which the patches were tracked, as for these tests we are only interested in the effect that the number of defined energy terms has on the probabilities that the importance sampling distributions attach to different transformations between consecutive frames. This is an effect that can be measured without regard to the particular states that the particles are in in the first of each frame pair. Note also that not resampling means that the influence of the patch likelihood function L_t on the state of the particles was very weak. The only point at which it exerted any influence at all was during the calculation of the weighted averages of the particles that shared hash keys, which, as discussed in section 3.3.3, is only done so that the particles in each group can use a consistent set of grid points.

The parameters that we manually defined for all components of the particle filter are summarised in table 3.1, and the distributions of the number of subregions that defined each calculated sampling distribution are summarised with the percentiles given in table 3.2. None of the $\sim 321 \times 10^6$ energy terms that we calculated for 230108.FO1 were as large as the upper bound ψ^\uparrow , and only 0.009% of the $\sim 335 \times 10^6$ energy terms that we calculated for 180608.HS were $\geq \psi^\uparrow$, so the effect of energy capping on the conclusions we will draw from the tests is negligible.

Including the initial frames, we tracked the 4 patches over 20 frames in the 230108.FO1 sequence, and over 19 frames in 180608.HS (see figure 3.9 for examples of the tracked patch states from 5 frames of each sequence). As we do not cache the skewing, scaling and rotation sampling distributions, the number of times each of these three types of distribution was calculated is given by

$$(\text{no. patches}) \times (\text{no. particles}) \times (\text{no. frames distributions calculated over}) . \quad (3.55)$$

For 230108.FO1, this is $4 \times 50 \times (20 - 1) = 3800$, and for 180608.HS, this is $4 \times 50 \times (19 - 1) = 3600$. Note that this is not quite the number of distinct instances of each type of distribution however, since the particles were all in the same state in the initial frame. We have not checked the exact number of distinct instances, but it should be about $3800 - 49$ for 230108.FO1 and $3600 - 49$ for 180608.HS, since the probability of the event of two particles ending up in the exact same state after the initial frame (without resampling) may, for most practical purposes, be considered to be 0. There were less displacement sampling distributions however, due to caching. We calculated 2354 distinct displacement distributions for 230108.FO1, and 2610 for 180608.HS.

To justify the assumptions we made about the behaviour of the partial means, we considered two sets of empirical bivariate distributions constructed from the energy data for each video sequence. For each state x of a sampling distribution \tilde{f} , let $n_{\tilde{f}}(x)$ be the number of defined energy terms that the subregions contribute to $\tilde{f}(x)$. For each \tilde{f} and x , we uniformly sampled a random permutation $\boldsymbol{\rho}_{1:n_{\tilde{f}}(x)}$ over the subregions using the algorithm described in [31], and

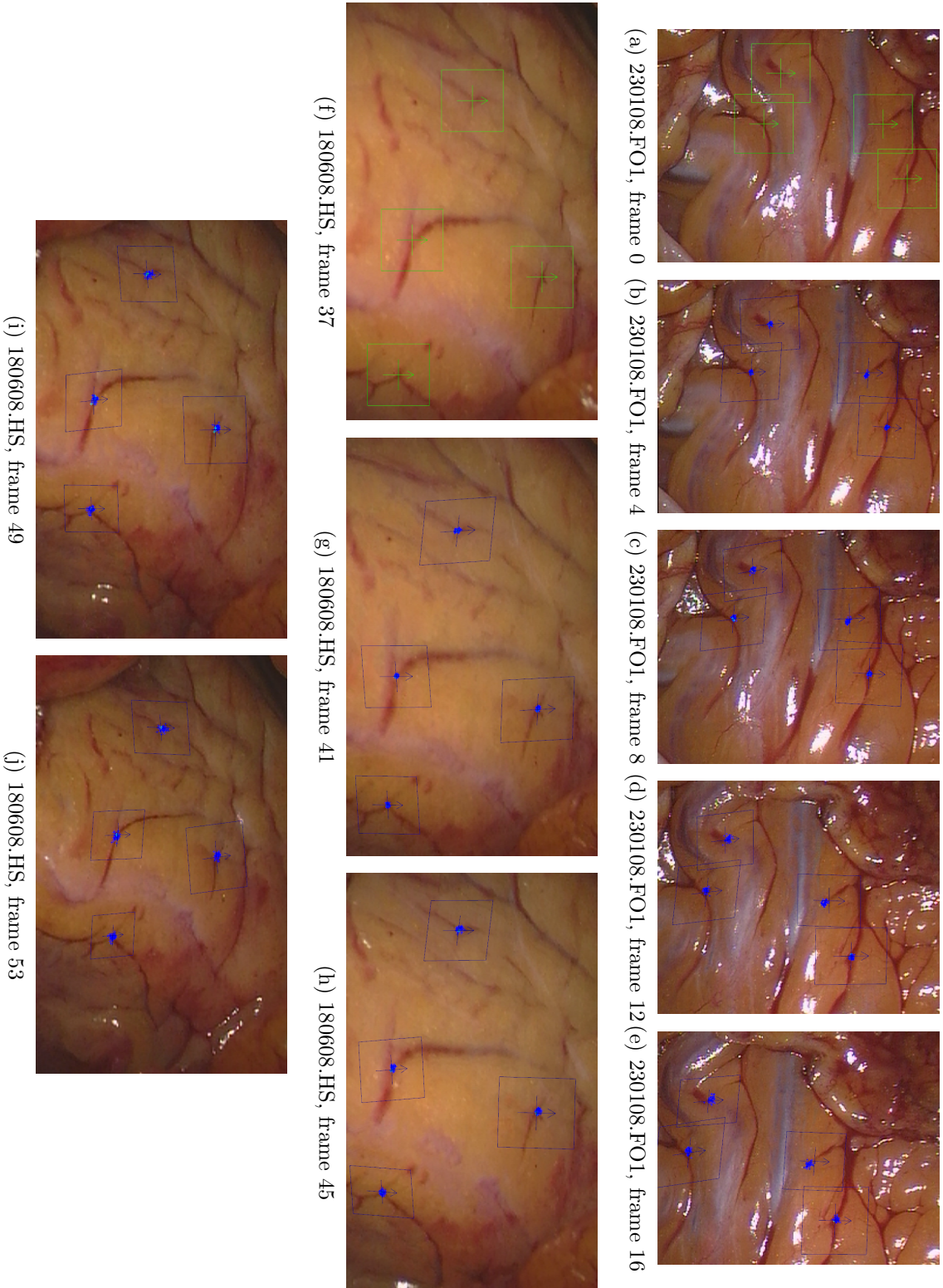


Figure 3.9: 5 frames from the 20/19 frames over which we tracked four patches in 230108.FO1/180608.HS. Each rendered patch is the weighted average of its particles. The green squares denote the manually-defined initial positions of the patches. The (mostly blue) clouds of crosses around the centres of each patch denote the positions of the particles.

we calculated all $n_{\tilde{f}}(x)$ partial means*. In a slight change to the previous section's notation, we will now use $\bar{\psi}_{x,i}$ to refer to the partial mean at state x of sampling distribution \tilde{f} , calculated using the first i defined energies contributed by subregions chosen in the order prescribed by $\rho_{1:n_{\tilde{f}}(x)}$. For the most part, we will omit $\bar{\psi}_{x,i}$'s dependence on \tilde{f} , to avoid notational clutter. But in cases where it is necessary to make this dependence explicit, we will write $\bar{\psi}_{\tilde{f},x,i}$.

The first set of empirical distributions we considered consisted of a distribution for each type of importance sampling distribution over the pairs

$$\left(\frac{i}{n_{\tilde{f}}(x)}, \max_{j \geq i} \frac{|\bar{\psi}_{x,j} - \bar{\psi}_{x,n_{\tilde{f}}(x)}|}{\bar{\psi}_{x,n_{\tilde{f}}(x)}} \right), \quad (3.56)$$

for each x and \tilde{f} of the given type, and all $i \in \{1, \dots, n_{\tilde{f}}(x)\}$. This set allows us to examine the validity of the assumption that the partial means are likely to converge as the number of energy terms increases, by providing data about how great the absolute relative error (ARE) in the estimates of the $n_{\tilde{f}}(x)^{\text{th}}$ partial mean may still be when no less than i energy terms are used.

The histograms that resulted from calculating these pairs from the partial means for the displacement, rotation, scaling and skewing sampling distributions for both test videos are shown in subfigures (a), (c), (i) and (k) of figures 3.11 and 3.12. Beneath each subfigure, we have plotted conditional histogram percentiles, each of which was defined by restricting the pairs to those for which $\frac{i}{n_{\tilde{f}}(x)}$ fell within a specific bin (for all tests in this section, we partitioned the range of $\frac{i}{n_{\tilde{f}}(x)}$ – the unit interval $[0, 1]$ – into 100 bins). Let

$$i_{\tilde{f}}(x, u) \triangleq \lceil n_{\tilde{f}}(x)u \rceil \quad (3.57)$$

*For each state x , given the condition that exactly $n_{\tilde{f}}(x) - i$ energy terms are undefined, the i^{th} partial mean (which we take as an approximation of the $n_{\tilde{f}}(x)^{\text{th}}$) is a random variable that fluctuates according to which i of the $n_{\tilde{f}}(x)$ subregions contribute the defined energy terms. The most accurate way to simulate the process of drawing samples from the i^{th} partial mean's distribution would involve repeatedly drawing random samples of plausible occlusion masks from some appropriate distribution until exactly $n_{\tilde{f}}(x) - i$ energy terms are undefined for state x . As a simpler and significantly more efficient proxy for this process, we take the most conservative approach of assuming all possible configurations in which exactly i subregions contribute defined energy terms to be equally probable.

be the minimum number of energy terms that must have contributed towards a partial mean at state x of sampling distribution \tilde{f} when the proportion of used energy terms was at least $u \in [0, 1]$. The passing of the $100p^{\text{th}}$ percentile curve through a point (u, v) indicates that v is the supremum, over $100p\%$ of the upper partial mean subsequences $\bar{\psi}_{x, i_{\tilde{f}}(x, u):n_{\tilde{f}}(x)}$, of the AREs between each subsequence term $\bar{\psi}_{x, \cdot}$ and $\bar{\psi}_{x, n_{\tilde{f}}(x)}$.

As expected, the percentiles show that the proportion of data samples with AREs greater than any given value decreases as the minimum number of energy terms used to calculate the partial means tends to $n_{\tilde{f}}(x)$. For 230108.FO1, only 10% of the upper partial mean subsequences for the displacement distributions had AREs greater than 0.1 when $0.3n_{\tilde{f}}(x)$ energy terms were used. When $0.56n_{\tilde{f}}(x)$ energy terms were used, only 1% exceeded an ARE of 0.1, and 90% had AREs ≤ 0.06 . The convergence was faster for the scaling distributions, for which $0.51n_{\tilde{f}}(x)$ energy terms were sufficient to bring 99% of the partial means within an ARE of 0.1, and faster still for the rotation and skewing distributions, for which $0.44n_{\tilde{f}}(x)$ and $0.41n_{\tilde{f}}(x)$ energy terms sufficed respectively. The results for 180608.HS were similar.

The fact that the percentiles did not converge at similar rates across the four types of sampling distribution may be a result of the large differences in the sizes of the domains of each of these sampling distributions, and hence in the number of partial means that each type of sampling distribution contributed to the data sets. All of the bounded-displacement energy functions, and hence the displacement sampling distributions, were sampled at $(2^{\frac{d^*}{\delta_d}} + 1)^2 = 1681$ (under the parameter values in table 3.1) points. The exact numbers of points at which the other three sampling distribution types were sampled varied, due to the automatic selection of the sample spacing that we describe in appendix B.3, but the numbers were 1 order of magnitude less than 1681 for the scaling distributions, and 2 orders less for the skewing and rotation distributions.

For each sampling distribution \tilde{f} , let $x_{\tilde{f}}^{\downarrow}$ denote the state of minimal $n_{\tilde{f}}(x)^{\text{th}}$ partial mean, i.e.:

$$x_{\tilde{f}}^{\downarrow} \triangleq \arg \min_{x'} \bar{\psi}_{x', n_{\tilde{f}}(x')} \quad . \quad (3.58)$$

The second set of empirical distributions we considered consisted of four distributions (again, one for each type of importance sampling distribution) over the pairs

$$\left(\frac{i}{n_{\tilde{f}}(x)}, \frac{\min_{j \geq i} \bar{\psi}_{x,j} - \max_{j \geq i} \bar{\psi}_{x_{\tilde{f}}^{\downarrow},j}}{\max_{j \geq i} \bar{\psi}_{x_{\tilde{f}}^{\downarrow},j}} \right), \quad (3.59)$$

for all $x \neq x_{\tilde{f}}^{\downarrow}$ in each \tilde{f} 's domain, and $i \in \{1, \dots, \min(n_{\tilde{f}}(x), n_{\tilde{f}}(x_{\tilde{f}}^{\downarrow}))\}$. The construction of this set provides evidence for the correctness of eq. (3.49) when comparing the partial mean energies of each sampling distribution \tilde{f} 's states to those of $x_{\tilde{f}}^{\downarrow}$, as the second element of each pair is the relative margin that separates the partial means at $x_{\tilde{f}}^{\downarrow}$ from those at another state x of \tilde{f} 's domain, when the partial means at both states use at least i energy terms. The relative margin is defined as

$$\frac{\alpha'}{\max_{j \geq i} \bar{\psi}_{x_{\tilde{f}}^{\downarrow},j}}, \quad (3.60)$$

where α' is the greatest value for which

$$\bar{\psi}_{x,d} - \bar{\psi}_{x_{\tilde{f}}^{\downarrow},c} \geq \alpha', \quad \forall (c, d) \in \left\{ i, \dots, \min(n_{\tilde{f}}(x), n_{\tilde{f}}(x_{\tilde{f}}^{\downarrow})) \right\}^2 \quad (3.61)$$

(see figure 3.10). When $\alpha' > 0$, it can be taken as an estimate of the α variable used in eq. (3.44) that defined the minimum rate of exponential growth in sampling probability ratios. When $\alpha' \leq 0$, it suggests that either i is not a large enough number of energy terms to overcome the fluctuations in the ordering of the partial means at states x and $x_{\tilde{f}}^{\downarrow}$, or that the ratio between the sampling probabilities at these states will not grow exponentially.

The histograms for this set of empirical distributions are shown in subfigures (b), (d), (j) and (l) of figures 3.11 and 3.12, once again with the conditional histogram percentile curves plotted beneath them. This time, the passing of the $100p^{\text{th}}$ percentile curve through a point (u, v) indicates that for $100(1 - p)\%$ of the upper partial mean subsequences $\bar{\psi}_{x, i_{\tilde{f}}(x, u): n_{\tilde{f}}(x)}$, $x \neq x_{\tilde{f}}^{\downarrow}$, the infimum on the relative margins for which eq. (3.61) holds is v . The value of v should be > 0 for most subsequences when u is sufficiently high, since, under the evidence for the convergence of partial means provided by the first set of empirical distributions, there should be little variation in the partial means that use at least $i_{\tilde{f}}(x, u)$ energy terms. The larger the

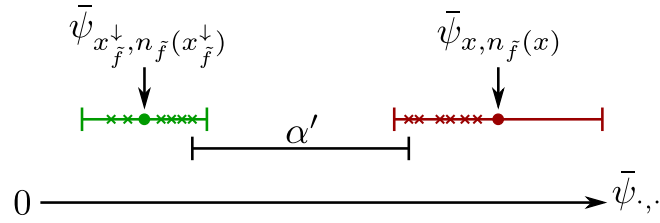


Figure 3.10: An illustration of the margin α' that separates the partial means $\bar{\psi}_{x,i:n}$ from the partial means $\bar{\psi}_{x_f^\downarrow,i:n}$, for $i = i_{\bar{f}}(x, u)$ (for some arbitrary u) and $n = \min(n_{\bar{f}}(x), n_{\bar{f}}(x_f^\downarrow))$. The latter partial means are indicated by the green markers on the left (the dot marks the $n_{\bar{f}}(x_f^\downarrow)^{\text{th}}$), and the former are indicated by the red markers on the right. The error bars on the two groups indicate the regions about the dots within which the other partial means must lie, as dictated by the 100th percentile curve of the ARE histograms. Increasing the value of u decreases the width of these bounds, and hence increases the margin α' .

proportion $(1 - p)$ of subsequences for which $v > 0$, the stronger the evidence is for a restricted version of eq. (3.49), in which the constraint $m'_a, m'_b \leq \min(n_{\bar{f}}(x), n_{\bar{f}}(x_f^\downarrow))$ is imposed on the consequent, and $a = x_f^\downarrow$, $b = x$, and $m' = i_{\bar{f}}(x, u) - 1$.

The percentile curves confirm that the relative margin generally does increase as u tends to 1. Reading the curves from right to left gives an empirical estimate of the rate at which the probability of the exponential sampling ratio growth hypothesis being true decreases with u , when comparing states to the state of minimal mean energy. E.g. for 230108.FO1's rotation distributions, 90% of the relative margins were non-negative when the partial means were defined with at least $0.9n_{\bar{f}}(x)$ energy terms, and this fell to 75% with at least $0.56n_{\bar{f}}(x)$ energy terms, and 50% with at least $0.18n_{\bar{f}}(x)$ energy terms.

In both videos, the percentile curves for the rotation, scaling and skewing distributions grew more slowly and covered narrower ranges than the percentile curves for displacement. E.g. for the displacement distributions for the 230108.FO1 video, 99% of the partial means using at least $0.16n_{\bar{f}}(x)$ energy terms had non-negative relative margins, and 90% had relative margins ≥ 0.2 when at least $0.3n_{\bar{f}}(x)$ energy terms were used. In comparison, the scaling distributions required at least $0.6n_{\bar{f}}(x)$ energy terms to be used in order for 90% of the partial means to have non-negative relative margins, and the rotation and skewing distributions required at least $0.9n_{\bar{f}}(x)$ and $0.94n_{\bar{f}}(x)$ energy terms, respectively. Again, these differences may be due to the large differences in the sizes of the domains of the different types of sampling distribution, which

meant that the domain of the displacement distributions had a higher proportion of states far away from x_f^\downarrow than the domains of the other types of distribution.

3.5 Conclusion

We have introduced a new importance sampling method inspired by Hinton’s Product-of-Experts model. By multiplicatively combining likelihood information from multiple subregions, our method constructs distributions from which the displacement, rotation, skewing and scaling components of a particle’s state can be sampled. We have empirically validated the hypothesis that, relative to the mode, the sampling probabilities of the vast majority of non-modal states in these distributions grow exponentially with the number of defined energy terms that the subregions contribute, and we have devised a simple method for handling missing likelihood data based on this hypothesis.

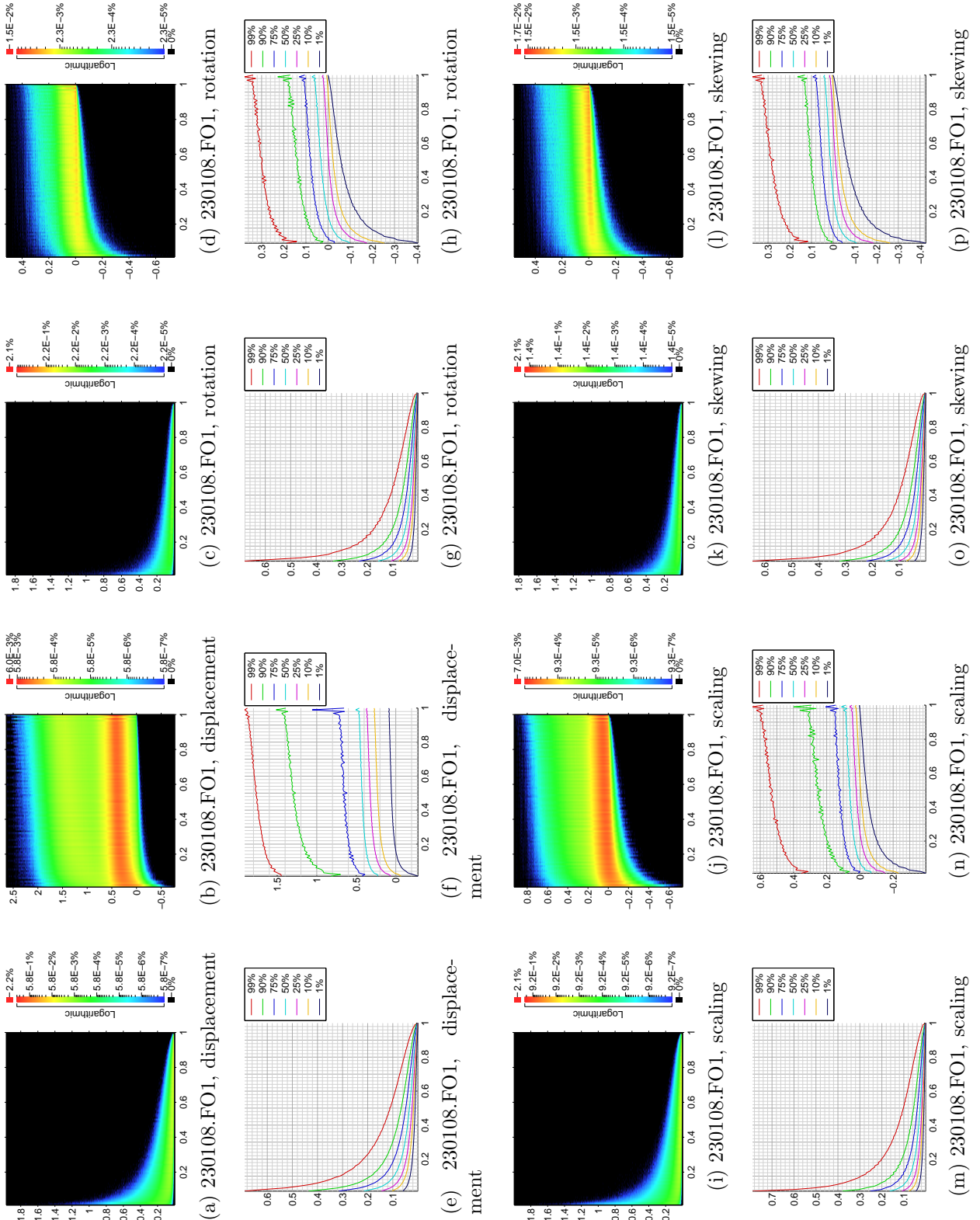


Figure 3.11: (a), (c), (i), (k) Histograms of the joint distributions of the pairs from eq. (3.56), for the 230108.FO1 video. In all subfigures, the x axis denotes values of $\frac{i}{n_{\tilde{f}}(x)}$. The y axes have been partitioned into 3000 bins, and the x axes into 100; we do this with all histograms in the rest of this section. The colour key follows a logarithmic scale, and no samples fell into the black regions. (b), (d), (j), (l) Histograms of the joint distributions of the pairs from eq. (3.59). (e)-(h), (m)-(p) The 1st, 10th, 25th, 50th, 75th, 90th and 99th percentiles of the histograms above them, under the condition that $\frac{i}{n_{\tilde{f}}(x)}$ lies in a specific bin.

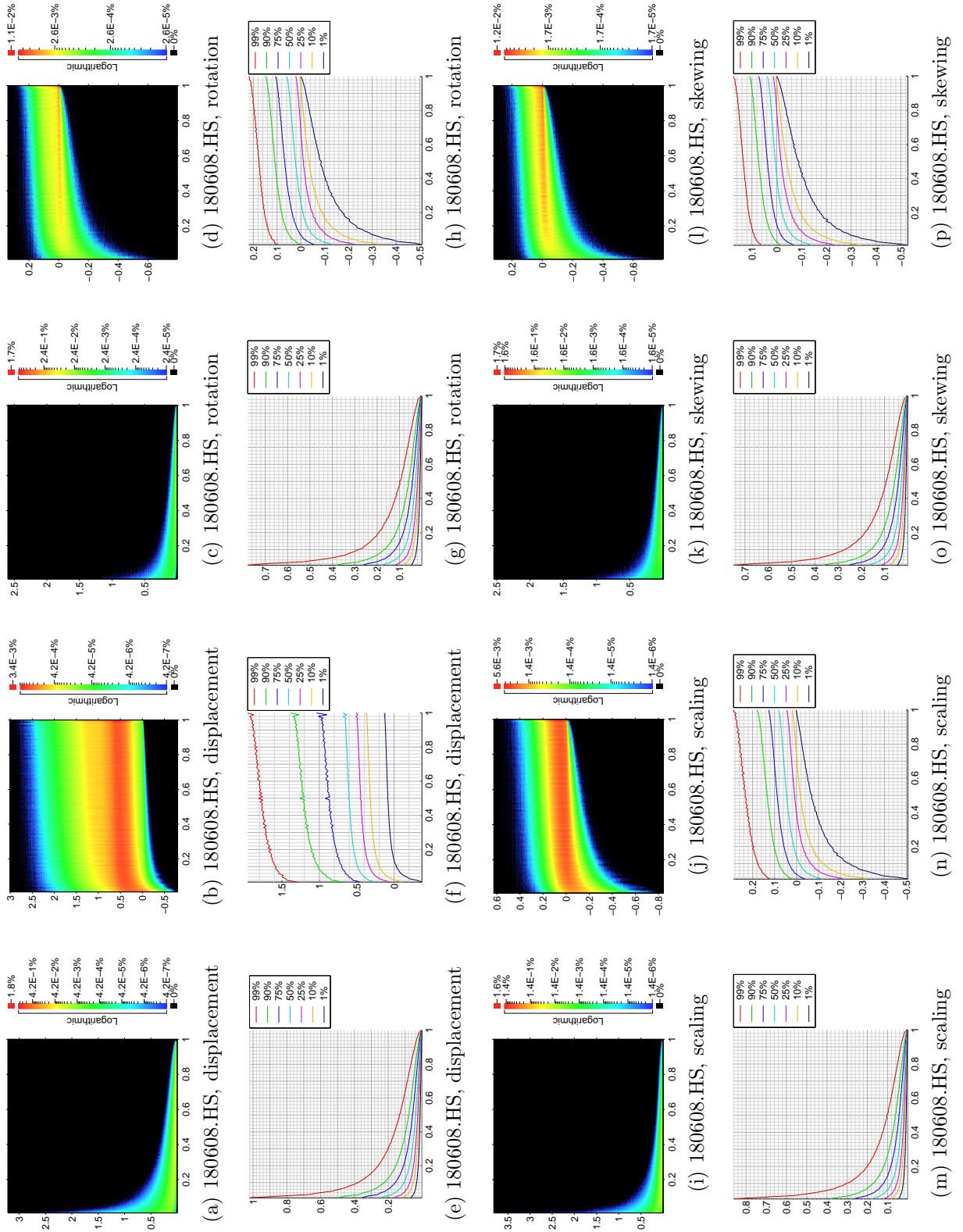


Figure 3.12: The figures here have a similar interpretation to those in figure 3.11, except that these figures are for the 180608.HS video.

Chapter 4

Likelihood Evaluation

4.1 Synopsis

In the previous chapter, we treated the function $\Delta_{\ell'}$, that defines the local likelihood energies, as a black box, and we did not touch on the form of the patch likelihood function L_t . For the particle filter to be successful, we must define these functions in a way that makes them invariant to changes in the appearance of a patch that are not caused by myocardial deformations, and insensitive to appearance changes caused by the non-affine components of myocardial deformations (since our importance sampler currently only hypothesises affine deformations).

The most significant non-deformational appearance changes that we must take into account are those caused by specular reflections and illumination changes. We handle these by carrying out two different image processing operations. For the specular highlights, we mask out and ignore regions of high intensity and low saturation. We remove the effects of illumination changes by fitting a parametric surface with few degrees of freedom to the lightness channel of an LUV colour space representation of the image, and then subtracting the surface from the channel. This has the effect of preserving fine textural details while attenuating lower-frequency shading-related lightness variations.

After applying these operations, we then compare patches based on the part of the (weighted)

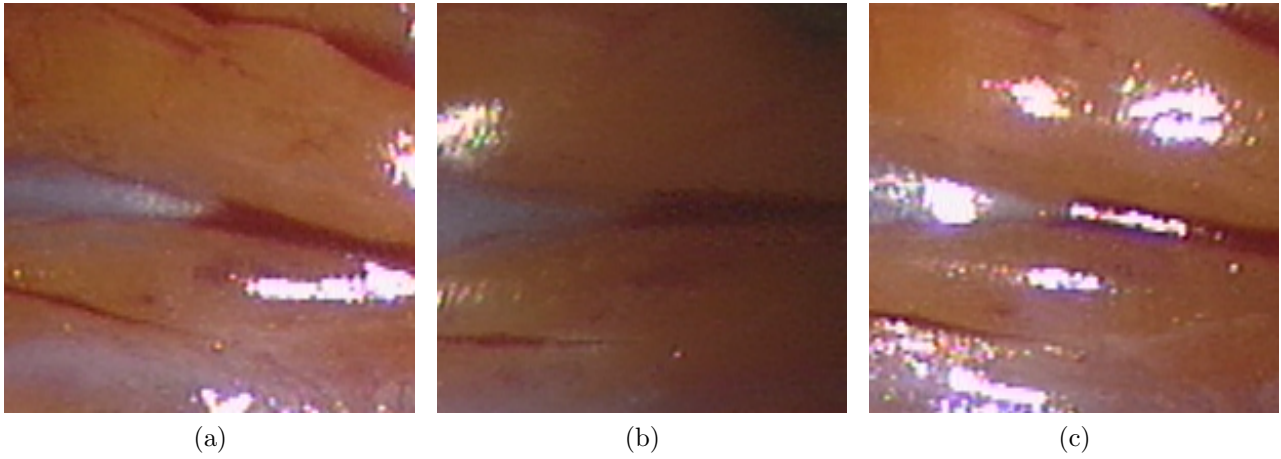


Figure 4.1: Three images showing the kinds of changes in the appearance of an image region that the dissimilarity functions must be able to cope with. (a) The original region. (b) An illumination change. (c) Occlusion by specular highlights.

distribution of squared differences between corresponding pixels that lies below a certain percentile. By discarding the part of the distribution that lies above the percentile, we hope to introduce a degree of insensitivity to the non-affine component of the deformations, and to provide extra resilience against specular highlights that are not detected by the mask.

4.2 Robust Image Region Comparison

As with ℓ_t , we define L_t in terms of a non-negative dissimilarity function $\Delta_{L,s,t}[\mathbf{V}(\theta_s), \mathbf{V}(\theta_t)]$, that compares the appearance of patch $\mathbf{V}(\theta_s)$ in image \mathbf{Z}_s to the appearance of patch $\mathbf{V}(\theta_t)$ in image \mathbf{Z}_t , and we model L_t as a stretched exponential function of it:

$$L_t(\Theta_t; \Theta_0, \mathbf{Z}_{[0,t]}) \propto \varphi(\Delta_{L,0,t}[\mathbf{V}(\Theta_0), \mathbf{V}(\Theta_t)]; \alpha_L, \gamma_L) , \quad (4.1)$$

for hyperparameters α_L and γ_L that we will deal with the selection of in chapter 6 (the constant of proportionality is unimportant for now, as it disappears when the particle weights are normalised). For now we will just focus on the forms of the dissimilarity functions.

Although myocardial deformations are a significant cause of changes in the appearance of patches, they are certainly not the only kinds. Any model of the dissimilarity functions that

define the local likelihood functions and the patch likelihood function must address the same question: given the hypothesised trajectory of the patch up till time t , what is the probability that (some region of) image Z_t looks the way it does? In answering this question, the dissimilarity functions must take into account the kinds of non-deformational appearance changes illustrated in figure 4.1, namely: changes in illumination/shading and partial or total occlusion by specular highlights (and possibly other objects, such as surgical tools). To cope with these phenomena, we define the dissimilarity functions in terms of two image filters:

- A binary occlusion mask $\mathcal{F}_{\mathcal{O},t}(\mathbf{x})$, that attempts to label each pixel \mathbf{x} that lies in an occluded region of Z_t with a value 1, and all other pixels with a value 0.
- A lightness-normalising filter $\mathcal{F}_{\mathcal{L},t}(\mathbf{x})$, that attempts to produce a representation of each pixel \mathbf{x} of Z_t that is invariant to changes in shading and illumination.

Given these filters, we define both $\Delta_{\ell'}$ and Δ_L in terms of a function $\mathcal{E}_{s,t}$ that measures the dissimilarity between regions X_s and X_t of Z_s and Z_t respectively, where the correspondence between points of X_s and points of X_t is determined by the hypothesised transformation. In particular, given a central reference point $\bar{\mathbf{x}}_s$ in X_s and a radial basis function $\omega(r)$ that we use to weight the contribution of each $\mathbf{x} \in X_s$, we want \mathcal{E}^2 to be a single number that summarises the weighted distribution of the squared Euclidean differences $d_{s,t}^2$ between all points $\mathbf{x} \in X_s$ and $\mathbf{x}' \in X_t$:

$$d_{s,t}^2(\mathbf{x}, \mathbf{x}') \triangleq \|\mathcal{F}_{\mathcal{L},s}(\mathbf{x}) - \mathcal{F}_{\mathcal{L},t}(\mathbf{x}')\|^2, \quad (4.2)$$

where each $(\mathbf{x}, \mathbf{x}')$ pair is associated with a weight $\omega'_{s,t}$:

$$\omega'_{s,t}(\mathbf{x}, \mathbf{x}') \triangleq (1 - \mathcal{F}_{\mathcal{O},s}(\mathbf{x})) (1 - \mathcal{F}_{\mathcal{O},t}(\mathbf{x}')) \omega_s(\mathbf{x}) \quad (4.3)$$

$$\omega_s(\mathbf{x}) \triangleq \omega(\|\mathbf{x} - \bar{\mathbf{x}}_s\|) . \quad (4.4)$$

The most obvious choice of \mathcal{E}^2 would be the weighted mean of $d_{s,t}^2$, evaluated over all pairs of corresponding points. However, it is well-known that the mean is sensitive to outliers, which can cause problems when $\mathcal{F}_{\mathcal{O}}$ is not able to mask out every pixel that has a strong specular

component, or when a significant minority of points in X_s near boundaries (such as vessel boundaries) look very different to the corresponding points in X_t due, not to the importance sampler hypothesising an incorrect state, but to the simplifying assumptions we make about the deformations of the patches and the local likelihood subregions, which mean that the best-fitting states will almost never perfectly register these regions. A more robust choice would be to consider the $100p^{\text{th}}$ percentile of the weighted distribution (e.g. the median, with $p = 0.5$). But a single percentile does not capture much information about the part of the distribution below it. An important feature of the distribution to consider is how close all of the mass below the $100p^{\text{th}}$ percentile is to 0, because a distribution with a large weighted proportion of sub-percentile $d_{s,t}^2$ values close to 0 should give a smaller value of \mathcal{E}^2 than a distribution with the same $100p^{\text{th}}$ percentile that attaches less weight to low values.

Let \mathbf{d}^2 be a random variable defined by sampling a pair of corresponding points $\mathbf{x} \in X_s$ and $\mathbf{x}' \in X_t$ with probability proportional to $\omega'_{s,t}(\mathbf{x}, \mathbf{x}')$, and setting $\mathbf{d}^2 = d_{s,t}^2(\mathbf{x}, \mathbf{x}')$. Also, let d_p^{*2} refer to the $100p^{\text{th}}$ percentile of \mathbf{d}^2 's distribution. \mathcal{E}^2 should return 0 if and only if the conditional random variable

$$\mathbf{d}'^2 \triangleq \mathbf{d}^2 \mid \mathbf{d}^2 \leq d_p^{*2} \quad (4.5)$$

follows the Dirac delta distribution $\delta(d)$, i.e. if $\mathbf{d}' = 0$ always. So a natural way to incorporate information into \mathcal{E}^2 about how far \mathbf{d}'^2 's distribution is from 0 is to measure the cost of transforming it into δ .

A useful measure of this cost is the Earth Mover's Distance (EMD, also known as the 1st Mallow's or Wasserstein distance). This function is a metric on the space of n -variate probability distributions that can be thought of as a measure of the minimum amount of energy involved in the process of turning one distribution into the other by "pushing" mass around, where the energy needed to move mass m over distance d is md (the name comes from the idea that the PDFs are like piles of earth). It has been used extensively in image retrieval, where its measure of the difference between histograms or more compact signatures of image regions often proves to be closer to the perceptual difference than the measure given by pointwise functions like L^p metrics and the Kullback-Leibler divergence (e.g. see [89]), although pointwise correspondence

is not typically assumed in such applications.

For continuous univariate random variables \mathbf{X} and \mathbf{Y} with CDFs $F(x)$ and $G(y)$ respectively, [64] states that the EMD $\varrho[F, G]$ satisfies

$$\varrho[F, G] = \int_{-\infty}^{\infty} |F(x) - G(x)| dx . \quad (4.6)$$

The CDF of $\delta(\cdot)$ is given by the Heaviside step function:

$$H_\delta(x) \triangleq \int_{-\infty}^x \delta(y) dy = \begin{cases} 0 & , x < 0 \\ 1 & , \text{otherwise.} \end{cases} \quad (4.7)$$

So if \mathbf{d}'^2 's CDF is H_p^* , then by Riemann-Stieltjes integration and a change of integration order:

$$\begin{aligned} \varrho[H_p^*, H_\delta] &= \int_0^{d_p^{*2}} 1 - H_p^*(x) dx \\ &= \int_0^{d_p^{*2}} \int_x^{d_p^{*2}} dH_p^*(y) dx \\ &= \int_0^{d_p^{*2}} \int_0^y dx dH_p^*(y) \\ &= \int_0^{d_p^{*2}} y dH_p^*(y) \\ &= E[\mathbf{d}'^2] \\ &= E[\mathbf{d}^2 \mid \mathbf{d}^2 \leq d_p^{*2}] . \end{aligned} \quad (4.8) \quad (4.9)$$

In other words, the EMD between δ and \mathbf{d}'^2 's distribution is equal to all of the following: the area of the region above H_p^* and below the line $y = 1$; \mathbf{d}'^2 's mean; the conditional mean of \mathbf{d}^2 , under the condition that it lies below the $100p^{\text{th}}$ percentile. So in the special case where $p = 1$, $\varrho[H_p^*, H_\delta]$ is \mathbf{d}^2 's mean, and in general it lies in the interval $[0, d_p^{*2}]$.

We define an idealisation \mathcal{E}^* of \mathcal{E} as

$$\mathcal{E}_{s,t}^*(X_s, X_t; \omega, p) \triangleq \begin{cases} \frac{1}{c} \sqrt{\varrho[H_p^*, H_\delta]} & , \frac{\int \omega'_{s,t}(\mathbf{x}, \mathbf{x}') d(\mathbf{x}, \mathbf{x}')}{\int \omega_s(\mathbf{x}) d\mathbf{x}} \geq \tau \\ \perp & , \text{otherwise} , \end{cases} \quad (4.10)$$

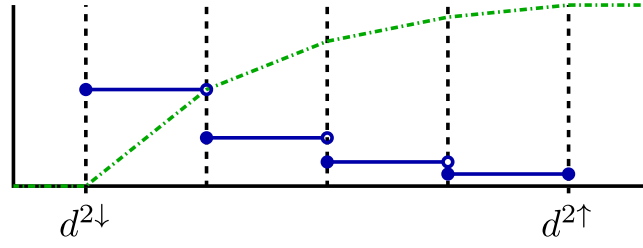


Figure 4.2: This figure illustrates the relationship between our definition of h (depicted by the blue horizontal line segments), the bin boundaries (filled circles indicate closed bounds, unfilled represent open bounds), and the CDF H_p (the green dot-dashed line segments). In this example, the mass of h over each bin and the gradient of H_p over each bin decrease by a factor of half from left to right. We approximate $\varrho[H_p^*, H_\delta]$ as $\approx \varrho[H_p, H_\delta]$, which is the area above H_p and below the line $y = 1$.

where: the integral in the numerator is taken over all pairs of corresponding points in X_s and X_t ; the integral in the denominator is taken over all points in X_s ; the scaling constant c ensures that \mathcal{E}^* 's image is the unit interval $[0, 1]$ (its value is the maximum possible Euclidean distance between pixel values, and so it only depends on the colour model that $\mathcal{F}_\mathcal{L}$ uses); and the threshold τ determines the maximum total weight that pixel pairs with an occluded pixel can have without \mathcal{E} failing and returning \perp , indicating that it is undefined. In all of the tests presented in this thesis, we use $\tau = 0.5$. Note that the square root does not necessarily have to be taken, as it can be incorporated into the value of φ 's α hyperparameter.

We cannot calculate $\varrho[H_p^*, H_\delta]$ exactly, as H_p^* and d_p^{*2} cannot be evaluated exactly in most practical cases. So instead, we consider an indexed subset $\mathbf{x}_s(i)$ of X_s , with corresponding points $\mathbf{x}_t(i) \in X_t$, and we define the squared differences $d^2(i)$ and weights $\omega'(i)$ of these pairs as

$$d^2(i) \triangleq d_{s,t}^2(\mathbf{x}_s(i), \mathbf{x}_t(i))$$

$$\omega'(i) \triangleq \omega'_{s,t}(\mathbf{x}_s(i), \mathbf{x}_t(i)) . \quad (4.11)$$

$$(4.12)$$

We then estimate H_p^* and d_p^{*2} by constructing an N bin histogram h over the observed $d^2(i)$.

Let

$$\begin{aligned} d^{2\downarrow} &\triangleq \min_i d^2(i) \\ d^{2\uparrow} &\triangleq \max_i d^2(i) . \end{aligned} \quad (4.13)$$

We define the width W of h 's bins as

$$W \triangleq \frac{d^{2\uparrow} - d^{2\downarrow}}{N} , \quad (4.14)$$

and the boundary $b(j)$ of the j^{th} bin of h as

$$b(j) \triangleq \begin{cases} [d^{2\downarrow} + (j-1)W, d^{2\downarrow} + jW) & , 1 \leq j < N \\ [d^{2\uparrow} - W, d^{2\uparrow}] & j = N \end{cases} . \quad (4.15)$$

We then construct h by adding $\omega'(i)$ to bin $\max\left(1, \left\lceil \frac{d^2(i) - d^{2\downarrow}}{W} \right\rceil\right)$ for each i , and normalising. Assuming the probability density of \mathbf{d}^2 over each bin $b(j)$ to be uniform and proportional to $h(j)$, our estimate d_p^2 of d_p^{*2} is given by

$$d_p^2 = d^{2\downarrow} + \left(k + \frac{p - \sum_{j=1}^k h(j)}{h(k+1)} \right) W , \quad (4.16)$$

where k is the largest integer such that

$$\sum_{j=0}^k h(j) \leq p < \sum_{j=0}^{k+1} h(j) . \quad (4.17)$$

We define $h(0) = 0$ and treat $h(N+1)$ as any finite value > 0 (that we will never need to use), so that $k = 0$ for $p < h(1)$ and $k = N$ for $p = 1$.

Our estimate of H_p^* is given by the piecewise differentiable conditional CDF H_p of our piecewise

uniform interpretation of h :

$$H_p(d^2) \triangleq \begin{cases} \frac{1}{p} \left(\sum_{j=1}^{\lfloor \iota(d^2) \rfloor} h(j) + (\iota(d^2) - \lfloor \iota(d^2) \rfloor) h(\lfloor \iota(d^2) \rfloor + 1) \right) & , d^2 \in [d^{2\downarrow}, d_p^2] \\ 0 & , d^2 < d^{2\downarrow} \\ 1 & , d^2 > d_p^2 \end{cases}$$

$$\iota(d^2) \triangleq \frac{d^2 - d^{2\downarrow}}{W} . \quad (4.18)$$

Defining $H(j)$ as the scaled cumulative histogram

$$H(j) \triangleq \frac{1}{p} \sum_{i=1}^j h(i) , \quad (4.19)$$

and noting that the area above H_p is just the sum of the areas of trapeziums (as illustrated in figure 4.2), we approximate $\varrho[H_p^*, H_\delta]$ as follows:

$$\begin{aligned} \varrho[H_p^*, H_\delta] &\approx \varrho[H_p, H_\delta] \\ &= \int_0^{d_p} 1 - H_p(x) dx \\ &= d^{2\downarrow} + \sum_{j=1}^k \int_{b(j)} 1 - H_p(x) dx + \int_{d^{2\downarrow} + kW}^{d_p} 1 - H_p(x) dx \\ &= d^{2\downarrow} + \sum_{j=1}^k \frac{W}{2} ((1 - H(j)) + (1 - H(j-1))) + \frac{W}{2} \frac{p - pH(k)}{h(k+1)} (1 - H(k)) \\ &= d^{2\downarrow} + \frac{W}{2} \left(2k - \sum_{j=1}^k (H(j) + H(j-1)) + p \frac{(1 - H(k))^2}{h(k+1)} \right) , \end{aligned} \quad (4.20)$$

and we approximate the integrals in the condition of eq. (4.10) as

$$\frac{\int \omega'_{s,t}(\mathbf{x}, \mathbf{x}') d(\mathbf{x}, \mathbf{x}')}{\int \omega_s(\mathbf{x}) d\mathbf{x}} \approx \frac{\sum_i \omega'(i)}{\sum_i \omega_s(\mathbf{x}_s(i))} . \quad (4.21)$$

Function CalcDeltaEMD and its helper function, given in appendix C, describe an algorithm for calculating $\varrho[H_p, H_\delta]$. It requires $|\mathbf{x}_s|$ iterations to calculate $d^2(\cdot)$, $d^{2\downarrow}$ and $d^{2\uparrow}$, an additional $|\mathbf{x}_s|$ iterations to calculate h (assuming that the time taken to initialise it to 0 is negligible),

and, in the worst case, an additional N iterations to calculate $\varrho[H_p, H_\delta]$ (note that we do not need to explicitly normalise h). The algorithm is no slower than the fastest algorithms for calculating d_p^2 (which must either sort the values first in $O(|\mathbf{x}_s| \log |\mathbf{x}_s|)$ time and select d_p^2 in $O(|\mathbf{x}_s|)$ time, or construct h and use it to approximate d_p^2), and only requires at most $|\mathbf{x}_s| + N$ more iterations than calculating the weighted mean.

Under normal operation, we define $\Delta_{\ell',t}[S(\mathbf{g}), S(\mathbf{g}) + \mathbf{d}]$ in terms of the region of $\mathcal{F}_{\mathcal{L},t-1}$ contained within subregion $S(\mathbf{g})$ and the region of $\mathcal{F}_{\mathcal{L},t}$ contained within the displaced subregion $S(\mathbf{g}) + \mathbf{d}$ as

$$\Delta_{\ell',t}[S(\mathbf{g}), S(\mathbf{g}) + \mathbf{d}] = \mathcal{E}_{t-1,t} \left(S(\mathbf{g}), S(\mathbf{g}) + \mathbf{d}; N \left(\cdot; 0, \frac{r_s}{3} \right), p \right), \quad (4.22)$$

for some p , assuming an arbitrary, but consistent, ordering over the subregions' pixels (e.g. a row-major ordering). We weight the pixels with the Gaussian radial basis function

$$N(r; 0, \sigma) \propto \exp \left\{ -\frac{r^2}{2\sigma^2} \right\} \quad (4.23)$$

(the constant of proportionality does not matter, because it disappears when the histogram h is normalised) to compensate for the fact that by uniformly displacing $S(\mathbf{g})$'s pixels by \mathbf{d} rather than transforming them exactly as prescribed by the hypothesised transformations, we introduce errors that will generally increase with each pixel's distance from $\bar{\mathbf{x}}_{t-1}$. As mentioned in the previous chapter, it is sometimes useful to compare $\mathcal{F}_{\mathcal{L},t}$ to earlier images; we will discuss this in section 6.5. Note that since both the subregion radius r_s and the positions of $S(\mathbf{g})$'s pixels relative to their mean are fixed, the Gaussian weights w_i will never change, so they can be precomputed and cached for later use.

For $\Delta_{L,0,t}[\mathbf{V}(\Theta_0), \mathbf{V}(\Theta_t)]$, we define \mathbf{x}_0 as the (arbitrarily ordered) list of pixels within the boundary of $\mathbf{V}(\Theta_0)$, and we calculate their corresponding bilinear interpolation coefficients $\alpha(i)$ and $\beta(i)$. We always define $\hat{\mathbf{V}}$ as a square and $\Theta_0 = \theta_0$ as a translation, so

$$\begin{aligned} \alpha(i) &= \frac{\mathbf{x}_{0,x}(i) - \mathbf{v}_x^\dagger}{s} \\ \beta(i) &= \frac{\mathbf{x}_{0,y}(i) - \mathbf{v}_y^\dagger}{s}, \end{aligned} \quad (4.24)$$

where \mathbf{v}^\downarrow is $\mathbf{V}(\Theta_0)$'s bottom-left vertex and s is the length of $\mathbf{V}(\Theta_0)$'s sides. Taking $\hat{\mathbf{V}}$'s vertices $\mathbf{v}_1, \dots, \mathbf{v}_4$ to be defined in anticlockwise order starting from the bottom-left vertex, we define \mathbf{x}_t in terms of the interpolation coefficients:

$$\begin{aligned} \mathbf{x}_t(i) = & \{(1 - \alpha(i))M(\mathbf{v}_4; \Theta_t) + \alpha(i)M(\mathbf{v}_3; \Theta_t)\} \beta(i) + \\ & \{(1 - \alpha(i))M(\mathbf{v}_1; \Theta_t) + \alpha(i)M(\mathbf{v}_2; \Theta_t)\} (1 - \beta(i)) , \end{aligned} \quad (4.25)$$

and Δ_L as

$$\Delta_{L,0,t}[\mathbf{V}(\Theta_0), \mathbf{V}(\Theta_t)] = \mathcal{E}_{0,t} \left(\mathbf{x}_0, \mathbf{x}_t; N \left(\cdot; 0, \frac{s}{6} \right), p \right) , \quad (4.26)$$

for some p , again using a Gaussian radial basis function to account for the fact that the errors in $\mathbf{V}(\Theta_t)$ under the best-fitting Θ_t will generally increase towards the patch's boundary (as before, the Gaussian weights do not change, so they can be precomputed and cached). The ordinates of \mathbf{x}_t 's vertices will often be non-integral, so we calculate points in \mathcal{F}_L by bilinearly interpolating:

$$\begin{aligned} \mathcal{F}_L(\mathbf{x}) = & \mathcal{F}_L(\lfloor \mathbf{x}_x \rfloor, \lfloor \mathbf{x}_y \rfloor)(\lfloor \mathbf{x}_x \rfloor + 1 - \mathbf{x}_x)(\lfloor \mathbf{x}_y \rfloor + 1 - \mathbf{x}_y) + \\ & \mathcal{F}_L(\lceil \mathbf{x}_x \rceil, \lfloor \mathbf{x}_y \rfloor)(\mathbf{x}_x - \lfloor \mathbf{x}_x \rfloor)(\lfloor \mathbf{x}_y \rfloor + 1 - \mathbf{x}_y) + \\ & \mathcal{F}_L(\lfloor \mathbf{x}_x \rfloor, \lceil \mathbf{x}_y \rceil)(\lfloor \mathbf{x}_x \rfloor + 1 - \mathbf{x}_x)(\mathbf{x}_y - \lfloor \mathbf{x}_y \rfloor) + \\ & \mathcal{F}_L(\lceil \mathbf{x}_x \rceil, \lceil \mathbf{x}_y \rceil)(\mathbf{x}_x - \lfloor \mathbf{x}_x \rfloor)(\mathbf{x}_y - \lfloor \mathbf{x}_y \rfloor) . \end{aligned} \quad (4.27)$$

Just as we needed a way of handling undefined values of Δ_ℓ , which we presented in the previous chapter, we also need a way of handling undefined values of Δ_L . Let J_d be the set of particle indices such that $\Delta_{L,0,t}[\mathbf{V}(\Theta_0), \mathbf{V}(\Theta_t^{[j]})]$ is defined for each $j \in J_d$, and let J_u be the indices of the particles for which it is undefined. Without at least looking for relevant image information in earlier frames, we cannot assume that $\Theta_t^{[j]}$ is more likely than $\Theta_t^{[j']}$ for any $j, j' \in J_u$. So we once again take the most conservative approach, this time by maximising the entropy of the particle distribution $(\Theta_{0:t}^{[j]}, w_t^{[j]})$ by setting $L_t(\Theta_t^{[j]})$ to a constant $e^{-\psi_u}$ for each $j \in J_u$, where, as

we show in appendix B.5, the value of ψ_u given by

$$\begin{aligned}
 \psi_u &= \max \left(\frac{b}{d} - \frac{a}{c}, 0 \right) \\
 a &= \sum_{j' \in J_d} w'_t{}^{[j']} \ln \left(w'_t{}^{[j']} \right) & b &= \sum_{j' \in J_u} w'_{t-1}{}^{[j']} p_t(\Theta_t^{[j']}) \ln \left(w'_{t-1}{}^{[j']} p_t(\Theta_t^{[j']}) \right) \\
 c &= \sum_{j' \in J_d} w'_t{}^{[j']} & d &= \sum_{j' \in J_u} w'_{t-1}{}^{[j']} p_t(\Theta_t^{[j']})
 \end{aligned} \tag{4.28}$$

maximises the entropy, assuming the form of the unnormalised weight update equation given in eq. (3.16). Note that this result assumes that the maximum likelihood that L_t can assign to a particle is 1. If this is not the case, then the unnormalised weights $w'_t{}^{[j']}$ should be divided by the maximum likelihood before calculating ψ_u . Of course, this rescaling has no effect once the weights are normalised.

All that remains now is for us to discuss the construction of the filters $\mathcal{F}_{\mathcal{L}}$ and $\mathcal{F}_{\mathcal{O}}$. We will do this in the following sections.

4.3 Lightness Normalisation Filtering

In analysing and removing the effects of shading and illumination changes on an image, it is vital for us to be able to separate information about the brightness of each pixel from information about its colour. Furthermore, given our use of the Euclidean difference to compare pairs of pixels, it would also be ideal to process the images in a perceptually uniform colour space, i.e. a space in which pairs of pixels that we perceive to be equally dissimilar have equal Euclidean differences. The CIE LAB and CIE LUV colour spaces [93] provide adequate solutions to these issues. Both colour spaces approximate perceptual uniformity over three channels, one of which is a lightness channel, L^* , that measures the perceived brightness of a pixel. So before defining $\mathcal{F}_{\mathcal{L},t}$, we convert the input image \mathbf{Z}_t into either of these spaces (the conversion processes are described in [93]). We then carry out the remaining steps of removing shading and illumination

effects on the L^* channel alone.

4.3.1 Illumination Model

The graphics literature is replete with lighting models of varying degrees of physical accuracy. A common feature amongst many of them is the modelling of the radiance of the light reflected from a surface as a linear combination of diffuse radiance terms and specular radiance terms. Shafer describes the reflection of these distinct forms of light from a number of materials in [97], stating that specular reflection occurs when light incident on an object is perfectly reflected about the surface normal at the interface between the object's body and the air, as a result of the differing refractive indices of the object and the air, whilst diffuse reflection is a result of the incident light penetrating the interface and interacting with colourants within the object's body before being selectively absorbed/retransmitted (this process is illustrated in figure 4.3a). The Dichromatic Reflection Model that he proposes in that paper is a useful reference for the assumptions we will make in defining $\mathcal{F}_{\mathcal{L},t}$, because it has a simple mathematical statement, and yet it is quite general in the sense that it subsumes lighting models such as the Phong model [79], and can explain more physically accurate models such as the Blinn-Phong model [13] and Cook-Torrance model [21] up to Fresnel effects* (the physical accuracy of these and other models was studied in [72], where the Cook-Torrance model was shown to be one of the most accurate models under investigation).

The illumination model we assume is a modification of the Dichromatic Reflection Model. Given an arbitrary parameterisation of the myocardial surface under which we represent each point with the parameter vector \mathbf{p} , we express the total radiance r_t reflected from the 3D surface point that \mathbf{p} corresponds to at time t as a function of: wavelength λ ; the angle ϕ_i between the surface normal $\hat{\mathbf{n}}$ at the point and the direction $\hat{\mathbf{i}}$ of the incident light; the angle ϕ_v between $\hat{\mathbf{n}}$

*By "Fresnel effects" we mean reflection governed by the laws of the reflection and transmission of electromagnetic waves at an interface set out by the physicist Augustin-Jean Fresnel. In particular, these equations describe an interdependence in the effects of wavelength and angle of incidence on reflectance. The Dichromatic Reflection Model makes the simplifying assumption that wavelength and angle of incidence have independent effects on reflectance, but Shafer estimates the error introduced by this assumption to be small when the angle of incidence is no greater than 60° , which it is in most parts of the intraoperative videos in our application.

and the viewing direction $\hat{\mathbf{v}}$; and the angle ϕ_g between $\hat{\mathbf{i}}$ and $\hat{\mathbf{v}}$ (see figure 4.3b); as follows:

$$r_t(\lambda, \phi_i, \phi_v, \phi_g, \mathbf{p}) = m_s(\phi_i, \phi_v, \phi_g) \Phi_s(\lambda) + m_d(\phi_i, \phi_v, \phi_g) \Phi_d(\lambda, \mathbf{p}) , \quad (4.29)$$

where Φ_s and Φ_d are the spectral power distributions (SPDs) of the specular and diffuse components respectively, and m_s and m_d are scaling factors that only depend on the angles. Note that the photometric angles should all be considered to be functions of \mathbf{p} and t , but we suppress this to avoid notational clutter.

The most significant modification we have made is to allow Φ_d to vary spatially (in the original model, it was assumed to be fixed everywhere up to a scaling factor). Permitting spatial variations allows the *albedo* of the surface (an intrinsic surface property that controls the amount of incident light diffusely reflected from each surface point at each wavelength, and that hence determines the surface's texture) to be non-uniform. Three convenient assumptions from the original model that we do assume, however, are that:

1. there is a single light source (this is true in our case, and in fact $\hat{\mathbf{i}} \approx \hat{\mathbf{v}}$, since the light source is fixed to the endoscope),
2. the *relative* SPD (i.e. the distribution that results from scaling the SPD so that it integrates to 1) of the incident light is spatiotemporally invariant,
3. there are no interreflections between surfaces.

From these assumptions, it follows that

$$\Phi_d(\lambda, \mathbf{p}) = \rho(\lambda, \mathbf{p}) \Phi_i(\lambda) \varsigma_t(\mathbf{p}) , \quad (4.30)$$

where: $\rho(\lambda, \mathbf{p}) \in [0, 1]$ is the albedo at point \mathbf{p} ; Φ_i is the relative SPD of the light source; and $\varsigma_t(\mathbf{p})$ is a spatiotemporally varying scaling factor that accounts for the total radiance of the light over all wavelengths, shadowing and any attenuation in light intensity that may occur due to the inverse square law. Since the albedo is a time-invariant intrinsic surface property, an

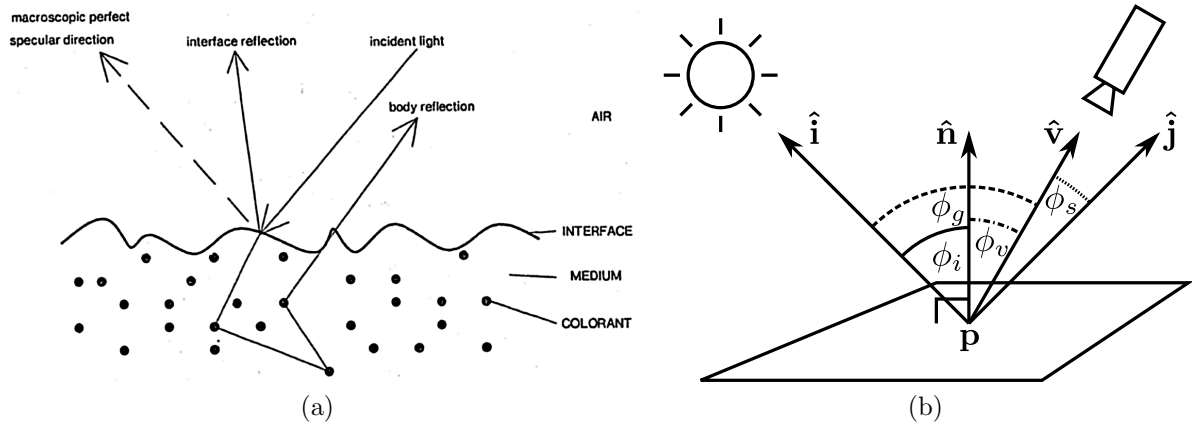


Figure 4.3: (a) The interactions between light and an inhomogeneous material. (b) The photometric angles used in the Dichromatic Reflection Model. Image (a) taken, and image (b) adapted, from [97].

estimate of it or of some time-invariant function of the triplet $(\lambda, \mathbf{p}, \rho(\lambda, \mathbf{p}))$ (we will explain what we mean by this in section 4.3.3) would be an ideal representation of L^* in the absence of illumination effects.

In addition to these three assumptions, we also assume that the camera’s exposure time is negligibly small, and hence that there is no significant motion blur.

4.3.2 Related Work

The estimation and/or removal of illumination effects is an important component of facial recognition/tracking algorithms, and a number of methods for solving the problem have been proposed in the literature. One group of related approaches replaces the task of estimating the parameters of a physics-inspired lighting model with that of estimating the coefficients of a linear model. For instance, [46] proposed a PCA-based approach in which the appearance of faces under arbitrary lighting conditions is modelled as a linear combination of “eigenfaces”. Similarly, [98] later showed that under perfect Lambertian reflection (for which $\Phi_s = 0$ and $m_d(\phi_i, \phi_v, \phi_g) = \cos \phi_i$) without shadows, three images of a surface taken from a fixed viewpoint, each with linearly independent light source directions, suffice as a linear basis that can be used to synthesise images from the same viewpoint under novel lighting conditions. [45] built on this result, asserting that the effects of shadowing can be approximately incorporated by

constructing a higher-dimensional linear basis. They demonstrated the effectiveness of this idea by incorporating the estimation of the basis coefficients into an optimisation-based tracker that calculates both illumination changes and affine shape changes in videos of human faces.

Unfortunately, all of these methods require training sets containing images of the object of interest under different lighting conditions, and so they would only be practical for the task of tracking arbitrary myocardial patches if we had a way of synthesising the appearance of the patches under randomly selected viewing conditions. This would only be possible if depth information was readily available for each image. If depth information was available however, we could define the camera to be at the origin and define $\hat{\mathbf{i}} \approx \hat{\mathbf{v}} = (0, 0, 1)^T$, and then calculate the surface normals at each point relative to this coordinate system with little difficulty. Under an appropriate model of m_d and m_s , this information would significantly simplify the problem of estimating the albedo.

Depth information could potentially be estimated for each image using shape-from-shading techniques. One such method, developed for an endoscopic stomach surgery application, was presented in [76]. It replaces many of the assumptions commonly made in shape-from-shading algorithms (a distant light source, orthographic projection and Lambertian reflection) with assumptions that are more appropriate for endoscopic images, such as ours. Unfortunately, it does not relax the assumption of a uniform albedo, and so it is not clear how well it would perform on our data. Shape-from-shading algorithms have been formulated under the assumption of non-uniform albedo however, and one such algorithm is given in [86], where, under the assumption that the surface can be locally approximated by planar patches, the authors express the problem in the form of an albedo-independent linear relationship between the image gradients (normalised by image intensity) and the patch's normal. We may give further attention to shape-from-shading techniques in future developments if more significant uses for depth information present themselves, but we felt that the present problem could be solved with simpler, more well-behaved image processing techniques.

One final common solution to the problems caused by illumination changes that we should mention is that of circumventing the problem altogether by working with image edges. Isard

and Blake’s particle filtering work was based on the idea of measuring the distances from image edges to the edges of a 2D line model [52], and more recent work has extended the idea to using 3D line models to track rigid 3D camera motions [83, 59]. The problem with using edges is that they discard a large amount of information about subtle textural variations in the interior of the segments they bound. Typical intraoperative myocardial images tend to be much richer in textural detail than they are in strong edges, so it is preferable to use an illumination change solution that preserves as much textural information as possible.

4.3.3 Local Lightness Normalisation

The fact that the myocardium is shiny means that the specular reflection term m_s generally produces specular highlights with sharp boundaries, beyond which the diffuse component is the dominant term of r . So for any point \mathbf{p} not within or near the boundary of a specular highlight, we assume

$$r_t(\lambda, \phi_i, \phi_v, \phi_g, \mathbf{p}) \approx m_d(\phi_i, \phi_v, \phi_g) \rho(\lambda, \mathbf{p}) \Phi_i(\lambda) \varsigma_t(\mathbf{p}) . \quad (4.31)$$

Let $C[x(\cdot)]$ be the functional that maps an SPD $x(\cdot)$ to the corresponding value of the L^* channel. As shown in [93], C is a strictly-increasing function of the CIE XYZ space’s Y channel (the channel that represents luminance), which in turn is defined as a linear combination of x and a fixed *colour matching function* (CMF). The mapping from Y to L^* is continuous everywhere and smooth at all but one point, at which a change is made from a linear model of our perception of dim light to a cubic root model of our perception of brighter light. Given this and the fact that for smooth surfaces like the myocardium, the diffuse scaling factor m_d is generally well modelled as a smoothly varying function of the photometric angles, it seems reasonable to model the local variations that m_d induces on regions of the L^* channel that correspond to a single surface using a parametric surface with a small number of degrees of freedom. By this we mean that if $\rho(\cdot, \mathbf{p})$ and $\varsigma_t(\mathbf{p})$ were spatially invariant, the mapping $\mathbf{q} \mapsto C[r_t(\cdot, \phi_i, \phi_v, \phi_g, \mathbf{p})]$, where \mathbf{q} denotes the image coordinate that corresponds to \mathbf{p} , would locally closely match some simple parametric model over the image coordinates \mathbf{q} . Noting that

any light intensity attenuation due to the inverse square law will also induce a smooth effect over regions of the L^* channel that correspond to a single surface, if we additionally assume that the boundaries of the shadows in each input image decay smoothly, then we can go one step further and use a simple parametric surface to model the local variations that $m_{d\varsigma_t}$ induces on L^* .

Let: $L'_{I,t}$ denote the time t input L^* channel; Q_t be a region of the channel corresponding to a single surface, over which m_s is negligible; $S'(\mathbf{q} \in Q_t; \boldsymbol{\mu})$ be a parametric surface that approximates the effect that $m_{d\varsigma_t}$ induces over this region of $L'_{I,t}$, where $\boldsymbol{\mu}$ is S' 's parameter vector; and P_t be the set of myocardial surface parameters that project into Q_t without being occluded. We will use $\mathbf{p} \in P_t$ to denote the surface parameter that maps to whatever $\mathbf{q} \in Q_t$ we happen to be talking about, notationally omitting \mathbf{p} 's dependence on \mathbf{q} for clarity.

As L^* is approximately perceptually uniform, we use an additive expression to approximately map the relationship of eq. (4.31) to L^* :

$$L'_{I,t}(\mathbf{q} \in Q_t) \approx S'(\mathbf{q}; \boldsymbol{\mu}) + \rho'(\mathbf{q}) , \quad (4.32)$$

for some function $\rho'(\mathbf{q})$ that represents the effects that ρ induces on $L'_{I,t}$. Noting that [93] gives the following expression for a function $C'(Y)$ that, up to a strictly-increasing linear function, maps the Y channel of CIE XYZ space to L^* :

$$C'(Y) = \begin{cases} \sqrt[3]{Y} & , Y > \left(\frac{24}{116}\right)^3 \\ \frac{841}{108}Y + \frac{16}{116} & , \text{ otherwise } , \end{cases} , \quad (4.33)$$

it follows that, up to the errors in S' , eq. (4.32) is equivalent to approximating $C'(Y)$ with a strictly-increasing linear function of $\log(Y)$, since

$$\begin{aligned} Y &= \int r_t(\lambda, \phi_i, \phi_v, \phi_g, \mathbf{p}) w(\lambda) d\lambda \\ &\approx m_d(\phi_i, \phi_v, \phi_g)_{\varsigma_t}(\mathbf{p}) E_{\Phi_i}[w(\lambda) \rho(\lambda, \mathbf{p})] \\ \Rightarrow \log(Y) &\approx \log(m_d(\phi_i, \phi_v, \phi_g)_{\varsigma_t}(\mathbf{p})) + \log(E_{\Phi_i}[w(\lambda) \rho(\lambda, \mathbf{p})]) , \end{aligned} \quad (4.34)$$

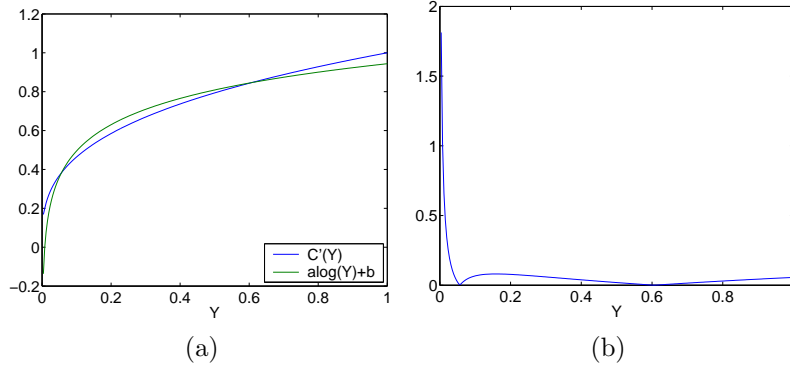


Figure 4.4: (a) A comparison of $C'(Y)$ and $a \log(Y) + b$. (b) A plot of the absolute relative error $\frac{|C'(Y) - a \log(Y) - b|}{C'(Y)}$.

where $w(\lambda)$ is the CMF for channel Y , so that for some variables a and b , and arbitrary $\alpha \in [0, 1]$:

$$\begin{aligned} S'(\mathbf{q}; \boldsymbol{\mu}) &\approx a \log(m_d(\phi_i, \phi_v, \phi_g)_{\varsigma_t}(\mathbf{p})) + \alpha b \\ \rho'(\mathbf{q}) &= a \log(E_{\Phi_i}[w(\lambda)\rho(\lambda, \mathbf{p})]) + (1 - \alpha)b . \end{aligned} \quad (4.35)$$

The benefit of this logarithmic approximation and our earlier assumption of Φ_i 's spatiotemporal invariance is that ρ' only varies with the albedo ρ , which simplifies the task of separating ρ 's effects from those of $m_d \varsigma_t$ in L^* space. Figure 4.4 shows the errors introduced by this approximation over $Y \in \{\frac{1}{256}, \frac{2}{256}, \dots, 1\}$, using the least-squares estimate of a and b . For $Y > 0.05$, the absolute relative error does not exceed 0.081, which seems acceptably small given all of our other approximations. For $Y < 0.05$, the error approaches infinity as Y approaches 0, but such dark pixels do not generally occur in our application, so this does not particularly matter*.

We define a *lightness normalising filter* to be any procedure that estimates $\boldsymbol{\mu}$ for each Q_t , so that $\mathcal{F}_{\mathcal{L},t}$'s L^* channel, which we will refer to as $L'_{O,t}$ (“O” for “Output”), can be computed with the approximately photometric-angle-invariant expression

$$L'_{O,t}(\mathbf{q} \in Q_t) = L'_{I,t}(\mathbf{q}) - S'(\mathbf{q}; \boldsymbol{\mu}) \approx \rho'(\mathbf{q}) . \quad (4.36)$$

*Note that when the first case of eq. (4.33) holds, we could equate $\log(L'_{I,t})$ with the RHS of eq. (4.32) instead, and then attempt to extract ρ' and finally exponentiate it. But the approximation we have proposed is more efficient, as it does not require the calculation of the logarithms or exponentials of any of the pixels.

The degree to which $L'_{O,t}$ is invariant to the photometric angles is mostly dependent on how well S' models the effects of $m_d\varsigma_t$ and how accurately $\boldsymbol{\mu}$ can be estimated, but some dependencies may also result from the error in our logarithmic approximation of C' . The use of this expression to remove ρ 's effects from $L'_{I,t}$ is an example of what we meant earlier when we spoke of calculating a time-invariant function of the triplet $(\lambda, \mathbf{p}, \rho(\lambda, \mathbf{p}))$, since for a good estimate of $\boldsymbol{\mu}$, $L'_{O,t}$ will mostly only depend on $\rho(\cdot, \mathbf{p})$.

The simplest case is, of course, when ρ is uniform over all $\mathbf{p} \in P_t$, which implies that

$$L'_{I,t}(\mathbf{q} \in Q_t) \approx S'(\mathbf{q}; \boldsymbol{\mu}) + k, \quad (4.37)$$

for some constant k . In this case, $m_d\varsigma_t$ is the only component of r_t that has a significant effect on $L'_{I,t}$, and so we can use the least-squares method to estimate the $\boldsymbol{\mu}$ under which S' , up to an underdetermined additive constant, best models this effect.

In the more general case of non-uniform ρ , suppose P_t could be segmented into regions over which ρ is approximately uniform. If the largest of these segments accounts for at least 50% of P_t , the remaining points of P_t could be treated as outliers, and so a robust least-squares method, such as RANSAC [34], could potentially be used to estimate $\boldsymbol{\mu}$ without having to explicitly calculate the segmentation. Note that if the largest segment corresponded to a vessel rather than the myocardium, this could cause S' to be uniformly offset relative to the surfaces that we fit to other regions. This is acceptable if the surfaces that model P_t are offset by the same amount for all t , as this will still lead to a representation of L^* that is invariant to illumination changes. We can usually ensure that the myocardium accounts for at least 50% of P_t by simply increasing P_t 's size, but making it too large will reduce the accuracy with which the chosen type of parametric surface models the effects of $m_d\varsigma_t$.

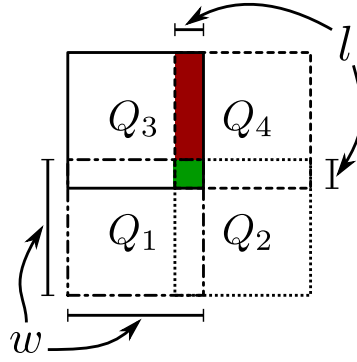


Figure 4.5: An illustration of the manner in which we arrange the squares over which we fit the surfaces. We define the overall value of the surface in regions covered by two squares, such as the red rectangle, by cubic interpolation. In regions covered by four squares, such as the green square, we use bicubic interpolation.

4.3.4 Parametric Surface Fitting

We considered two types of parametric surface in our tests: planar surfaces of the form

$$S_P(\mathbf{q}; \mathbf{a}) \triangleq (\mathbf{a}_{1:2})^T \mathbf{q} + \mathbf{a}_3 \quad , \quad (4.38)$$

for a 3D coefficient vector \mathbf{a} ; and circular paraboloids of the form

$$S_C(\mathbf{q}; \mathbf{b}) \triangleq \mathbf{b}_1(\mathbf{q}_x^2 + \mathbf{q}_y^2) + S_P(\mathbf{q}; \mathbf{b}_{2:4}) \quad , \quad (4.39)$$

for a 4D coefficient vector \mathbf{b} . We chose these forms for their simplicity, and because the least-squares parameter estimates for both surfaces have simple linear forms. Despite the sensitivity of the least-squares estimate to outliers, we found the results that we achieved using these two surface types and the methods we are about to describe adequate, even over regions of non-uniform albedo, so we did not use more robust fitting methods.

We began by dividing $L'_{I,t}$ into a regular grid of $w \times w$ overlapping squares, with an overlap of $l \leq \lfloor \frac{w}{2} \rfloor$ between all neighbours, as shown in figure 4.5. For each square Q , we fitted the surface to the pixels of the subset Q' that $\mathcal{F}_{\mathcal{O},t}$ labels as not being occluded. For any surface $f(\mathbf{q}; \mathbf{c})$ over Q satisfying the parametric linearity condition

$$f(\mathbf{q}; \mathbf{c}) = \mathbf{c}^T g(\mathbf{q}) \quad (4.40)$$

for an arbitrary vector-valued function g , the solution to the least-squares problem

$$\mathbf{c} = \arg \min_{\mathbf{c}'} \sum_{\mathbf{q} \in Q'} (f(\mathbf{q}; \mathbf{c}') - L'_{I,t}(\mathbf{q}))^2 \quad (4.41)$$

is given by the Moore-Penrose pseudoinverse [78]. We will make use of this result numerous times in this thesis, and we defer a discussion of some of its uses in a slightly more general context to section 5.3.2. For now we will just state the solution:

$$\mathbf{c}^T = \left(\sum_{\mathbf{q} \in Q'} L'_{I,t}(\mathbf{q}) g(\mathbf{q})^T \right) \left(\sum_{\mathbf{q} \in Q'} g(\mathbf{q}) g(\mathbf{q})^T \right)^{-1}. \quad (4.42)$$

For the planar surface S_P , we use

$$g(\mathbf{q}) = (\mathbf{q}_x, \mathbf{q}_y, 1)^T, \quad (4.43)$$

and for the circular paraboloid surface S_C , we use

$$g(\mathbf{q}) = (\mathbf{q}_x^2 + \mathbf{q}_y^2, \mathbf{q}_x, \mathbf{q}_y, 1)^T. \quad (4.44)$$

It would be trivial to replace S_C with the more general quadratic surface, in which the quadratic terms are weighted independently, or to use higher-degree polynomial surfaces. But more complex surfaces may cause the least-squares method to overfit $L'_{I,t}$, in the sense of modelling the effects of albedo variations on it. So such surfaces would only be appropriate when using robust fitting methods.

After fitting an S_P or S_C (we will use S_* to refer to either type) surface to each square Q , the final stage was to define S' over each Q . We set S' to the value of S_* in regions covered by a single S_* , and we (bi)cubically interpolate in regions where multiple S_* overlap using the cubic kernel

$$k(\alpha \in [0, 1]) \triangleq (3 - 2\alpha)\alpha^2, \quad (4.45)$$

which is one of the four basis functions used in cubic Hermite interpolation [7]. It is a partic-

ularly attractive interpolator to use because it is efficient to compute and it has the following properties:

$$\begin{aligned} k(0) &= 0 \quad , & k(1) &= 1 \quad , \\ k^{(1)}(0) &= 0 \quad , & k^{(1)}(1) &= 0 \quad , \\ k(\alpha \in [0, 1]) &\in [0, 1] \quad , & k(1 - \alpha) &= 1 - k(\alpha) \quad , \end{aligned} \quad (4.46)$$

where $k^{(1)}$ is its first derivative. The fact that its first derivatives are zero at the end points helps to smooth the transitions between overlapping S_* , and the last property describes the nature of its symmetry, which is such that whether: a) the kernel is reflected; b) the overlapping S_* are rotated by 180° ; or c) the overlapping S_* are reflected; the result of the interpolation will be the same up to a 180° rotation or reflection.

Let: Q_1, Q_2, Q_3 and Q_4 be four overlapping squares arranged as in figure 4.5; $S_{*,i}$ denote the S_* that has been fitted to Q_i ; $[x_1, x_2]$ be the x ordinates of the region of overlap between squares in the same row; $[y_1, y_2]$ be the y ordinates of the region of overlap between squares in the same column. We use k to define S' at a point \mathbf{q} that lies in the red or green region of figure 4.5 as follows (assuming that x ordinates increase from left to right, and y ordinates increase from bottom to top):

$$S'(\mathbf{q}) = L(\mathbf{q}) \left(1 - k \left(\frac{\mathbf{q}_x - x_1}{x_2 - x_1} \right) \right) + R(\mathbf{q}) k \left(\frac{\mathbf{q}_x - x_1}{x_2 - x_1} \right) \quad , \quad (4.47)$$

where

$$\begin{aligned} L(\mathbf{q}) &= \begin{cases} S_{*,1}(\mathbf{q}) & , \mathbf{q}_y < y_1 \\ S_{*,1}(\mathbf{q}) \left(1 - k \left(\frac{\mathbf{q}_y - y_1}{y_2 - y_1} \right) \right) + S_{*,3}(\mathbf{q}) k \left(\frac{\mathbf{q}_y - y_1}{y_2 - y_1} \right) & , \text{ otherwise} \end{cases} \\ R(\mathbf{q}) &= \begin{cases} S_{*,2}(\mathbf{q}) & , \mathbf{q}_y < y_1 \\ S_{*,2}(\mathbf{q}) \left(1 - k \left(\frac{\mathbf{q}_y - y_1}{y_2 - y_1} \right) \right) + S_{*,4}(\mathbf{q}) k \left(\frac{\mathbf{q}_y - y_1}{y_2 - y_1} \right) & , \text{ otherwise} \end{cases} . \end{aligned} \quad (4.48)$$

We define S' 's value in the other regions covered by two squares in a similar fashion.

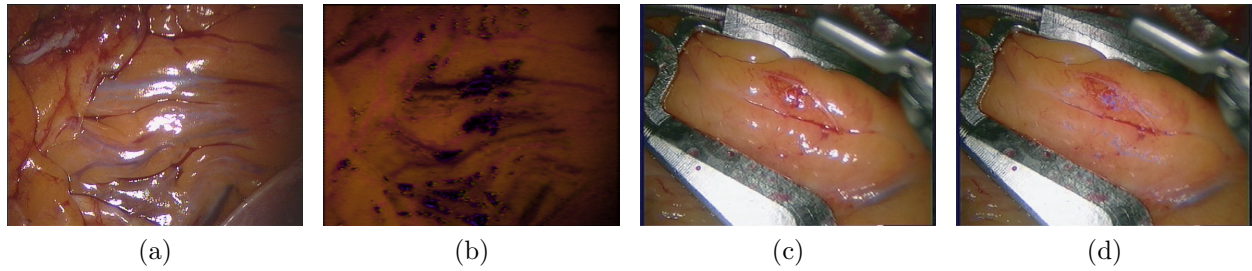


Figure 4.6: (a) An image from our 230108.FO1 sequence. (b) The result of applying Tan and Ikeuchi’s method [107] to (a) to remove the specular highlights, as computed by their source code. (c) An intraoperative image used by Stoyanov and Yang. (d) Stoyanov and Yang’s result after removing specular highlights from (c). Images (c) and (d) taken from [104].

4.4 Specular Highlight Masking

The two most common approaches to dealing with the distracting effects of specular highlights in computer vision problems are: 1) to detect and ignore them; 2) to detect and attempt to remove them, ideally leaving an image that only depends on the diffuse reflection components. Tan and Ikeuchi proposed an algorithm that follows the latter approach in [107]. Their method iterates between a process that checks whether or not the current estimate of the diffuse component still has specular components, based on the dichromatic reflection model and logarithmic differentiation, and a process that attempts to reduce the intensity of any remaining specular components, based on a transformation of the image pixels into a “maximum chromaticity intensity space”, in which the diffuse component of the specular pixels can, in theory, be identified from the intersection of two lines. Unfortunately, their method performs extremely poorly on our images, as shown in figures 4.6a and 4.6b. Its failure to restore the specular highlights is probably a result of the fact that they are so bright that they destroy the diffuse component information. Furthermore, their method has the undesirable effect of greatly reducing the contrast in regions where there are no specular highlights, which would only serve to make the tracking task more difficult than it already is.

In [103], Stoyanov and Yang proposed a modification of Tan and Ikeuchi’s method to improve its performance on intraoperative images. Their idea was to register multiple images from a video sequence together, so that the chromaticity information that bright specular highlights destroy can be recovered from other frames. Images of their results are shown in figures 4.6c and 4.6d.

The results are better, but there is still a lot of discolouration within the removed specular highlights. For our application, this discolouration would prove to be just as distracting as the specular highlights themselves, so we cannot adopt their approach either.

Unlike these approaches, we have chosen to handle specular highlights by detecting and ignoring them. We use an ad hoc method based on thresholding to construct specular highlight masks $\mathcal{F}_{\mathcal{O},t}$. Such methods are quite common in the image-guided surgery literature, and have, for example, been adopted by [90, 87].

In our implementation, we transform the video images into HSV space, and label pixels as specular if they have low saturation and high value (intensity). We calculate local thresholds over the saturation and value channels by partitioning the image into a regular grid of squares, calculating two 1D histograms over the saturation and value channels for each square, and defining the thresholds as percentiles of the histograms. This makes it easier to detect dim specular reflections without generating lots of false positives (i.e. labelling mostly-diffuse pixels as specular). We construct the histograms for a square S centred at \mathbf{p} over a larger square $S' \supset S$ also centred at \mathbf{p} , so as to improve labelling consistency across neighbouring squares. We typically define each S to be 101 pixels wide, and each S' to be 301 pixels wide. Percentile-based thresholds are not sufficient, however, as they may lead to false positives in squares that contain no specular highlights. So we also use spatially-invariant thresholds, labelling a pixel as specular if and only if its value and saturation are respectively greater than and less than both the percentile-based threshold and the spatially-invariant threshold. After completing the labelling, we finally dilate the mask a few times, to ensure that the boundaries of the specular highlights are sufficiently masked.

4.5 Results

4.5.1 Image Filtering

Figures 4.7 and 4.8 show typical results given by our lightness normalisation filtering and specular highlight masking methods. The effect of lightness normalisation is most pronounced in the second of these two figures, in which the visible part of the myocardium is more curved, leading to greater illumination-induced appearance variations. In the original images shown in this figure, the observable diffuse myocardial radiance is at its greatest (and hence, ignoring the specular highlights, the myocardium is at its lightest) in the area surrounding the largest region of specular highlights. Moving away from this specular region, the radiance decreases smoothly, eventually giving way to shadows in the top-left and bottom-right corners (the shadows are caused by the pericardial tissue that envelops the myocardium. Part of it can be seen hanging down on the left- and righthand sides of the images). All frames of this sequence were like this, since the camera and its light remained stationary, so the cardiorespiratory motion caused the vessels (which are usually the features that we are interested in tracking) to move back and forth in-between bright regions and darker regions. The lightness normalising filters help to remove the effects of these radiance variations, making the changes in the illumination of the vessels less pronounced. Note that some local lightness changes due to surface creases near vessels remain, e.g. near the top-left corner of the frame 62 images. These can be eliminated by using smaller grid squares, but making the grid squares too small reduces the contrast between the vessels and the myocardium.

By experimenting with the specular highlight mask thresholds, we were able to mask out most of the highlights, although a few dim ones remain undetected in the 230108.FO1 images. It is difficult (in fact, probably impossible) to find threshold values under which the mask will detect these without introducing more false positives, but even dim specular highlights may be sufficient to distract the importance sampler. One possible solution to this may be to compute “soft” masks, which assign a value in the interval $[0, 1]$ to each pixel, denoting an estimate of the proportion of its intensity that is caused by specular reflection. Such masks could be

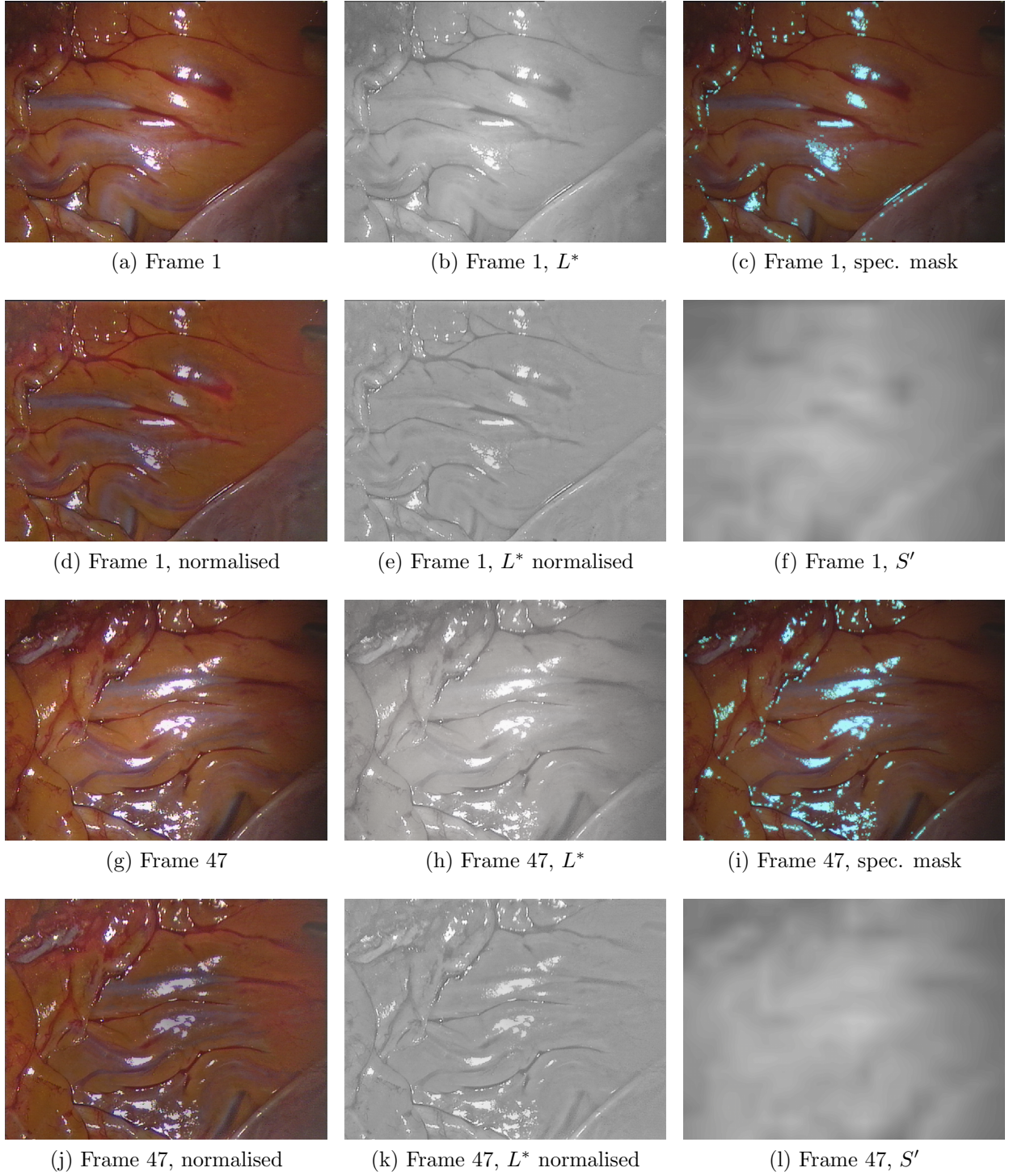


Figure 4.7: Examples of our lightness normalisation and specular highlight masking results for frames 1 and 47 of the 230108.FO1 sequence. For lightness normalisation, we used 60-pixel-wide planar surfaces with 30-pixel-wide overlaps (the image dimensions are 768×576). To facilitate comparison, we have added the mean lightness of frame 0 to all lightness normalised images. (a), (g) The original images from frames 1 and 47. (b), (h) The original L^* channels. (c), (i) Specular highlight masks – highlights shown in cyan. (d), (j) The result of applying lightness normalisation to the L^* channels and converting the images back to RGB space. (e), (k) The lightness normalised L^* channels. (f), (l) All of the S' surfaces used in lightness normalisation, rendered side by side.

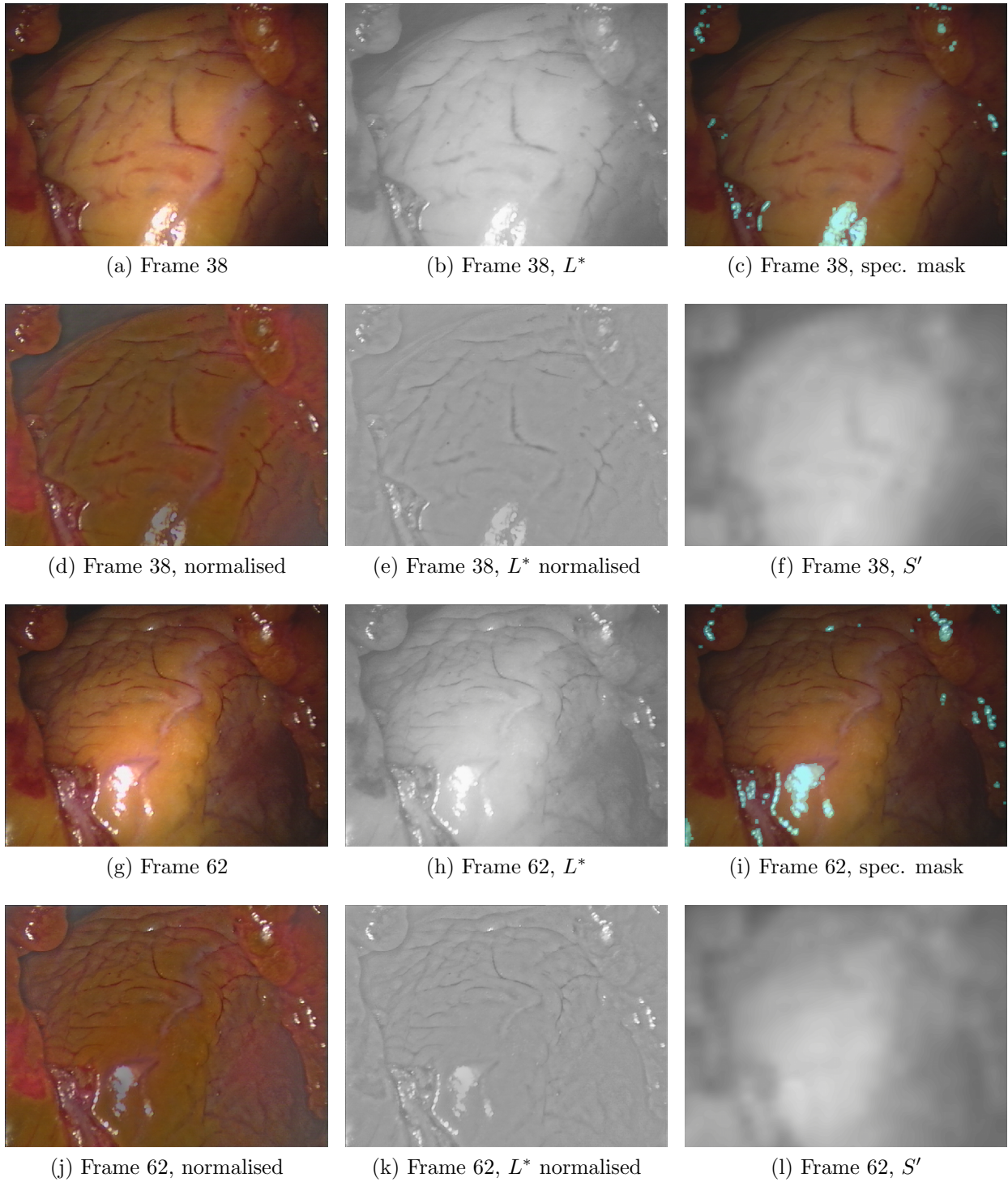


Figure 4.8: Examples of our lightness normalisation and specular highlight masking results for frames 38 and 62 of the 180608.HS sequence. All of the comments in the previous figure’s caption apply, except that we used circular parabolic surfaces in the lightness normalising filters this time.

used by our dissimilarity function \mathcal{E} without changing its definition, and may even simplify the problem of deciding what percentile to use with it.

The ideal choice of percentile is one that gives sufficient insensitivity to outliers without sacrificing \mathcal{E} 's ability to discriminate between states that are genuinely dissimilar. Our current binary masks produce outliers in the form of false negatives (mostly-specular pixels labelled as diffuse). Each false negative occurring at a pixel that, in the absence of specular reflection, would be considered similar to the pixel it is being compared to, reduces the number of outliers that \mathcal{E} can ignore under any given choice of percentile, i.e. reduces the extent to which it can ignore deformation-induced dissimilarities between similar regions. Soft masks would alleviate this problem by ensuring that false negatives are given low weight, so that significant outliers only occur at dissimilar diffuse pixel pairs. We intend to consider probabilistic approaches to soft mask construction in our future research.

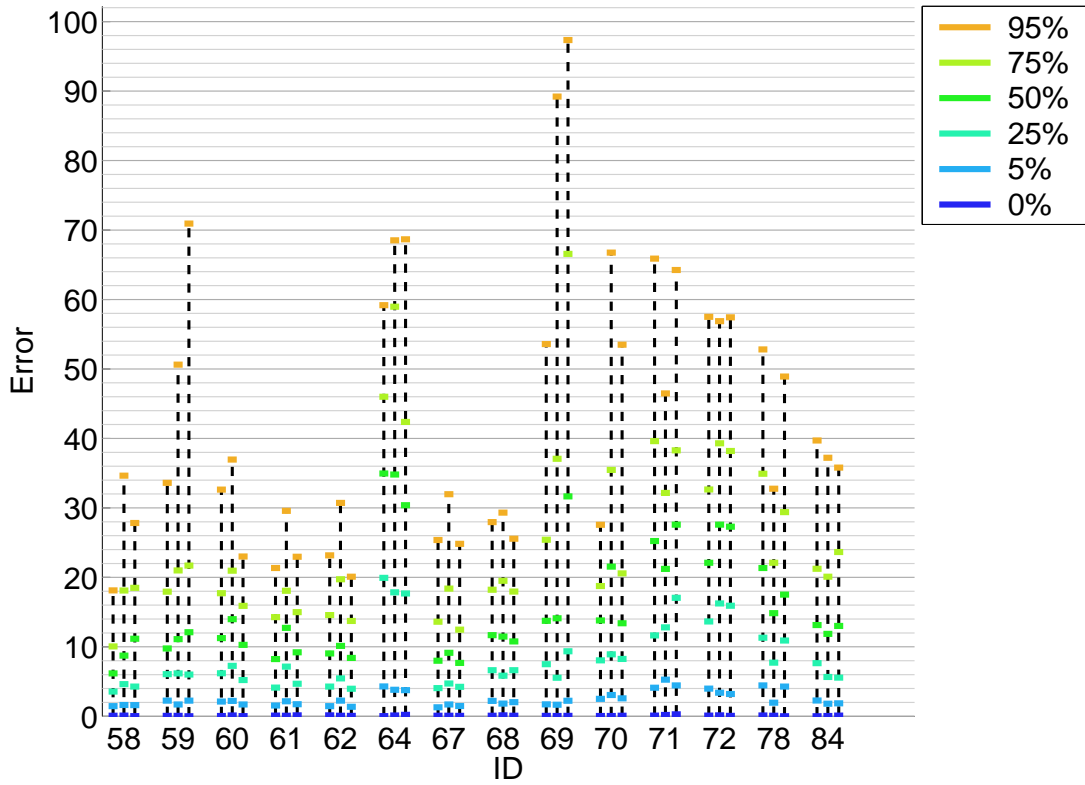
4.5.2 Comparative Tests

To test the extent to which our lightness normalisation and image comparison methods help to improve the performance of our particle filter, we tracked 14 patches in each video using the likelihood evaluation methods described in this chapter (for screenshots of the results, see section 6.6. The results section at the end of chapter 5 gives a statistical summary of the amount of motion/deformation in the two videos we used), and then we performed two comparative experiments. In one, we tried tracking the same patches without applying our lightness normalisation procedure to the L^* channel, and in the other, we did use lightness normalisation, but replaced our image region comparison function \mathcal{E} with the weighted sum of squared differences between the pixels. For the latter test, we used the same weighting method that we discussed in section 4.2, making this test equivalent to using \mathcal{E} with the percentile threshold p set to 1 (i.e. not discarding any part of the weighted pixel difference distributions). In all tests, we used the methods that we will discuss in chapter 6 to calculate maximum-likelihood estimates of the particle filter parameters (the kurtosis parameters α_L and α_{ℓ} , the inverse decay rate parameters γ_L and γ_{ℓ} , and two things that we will introduce in that chapter:

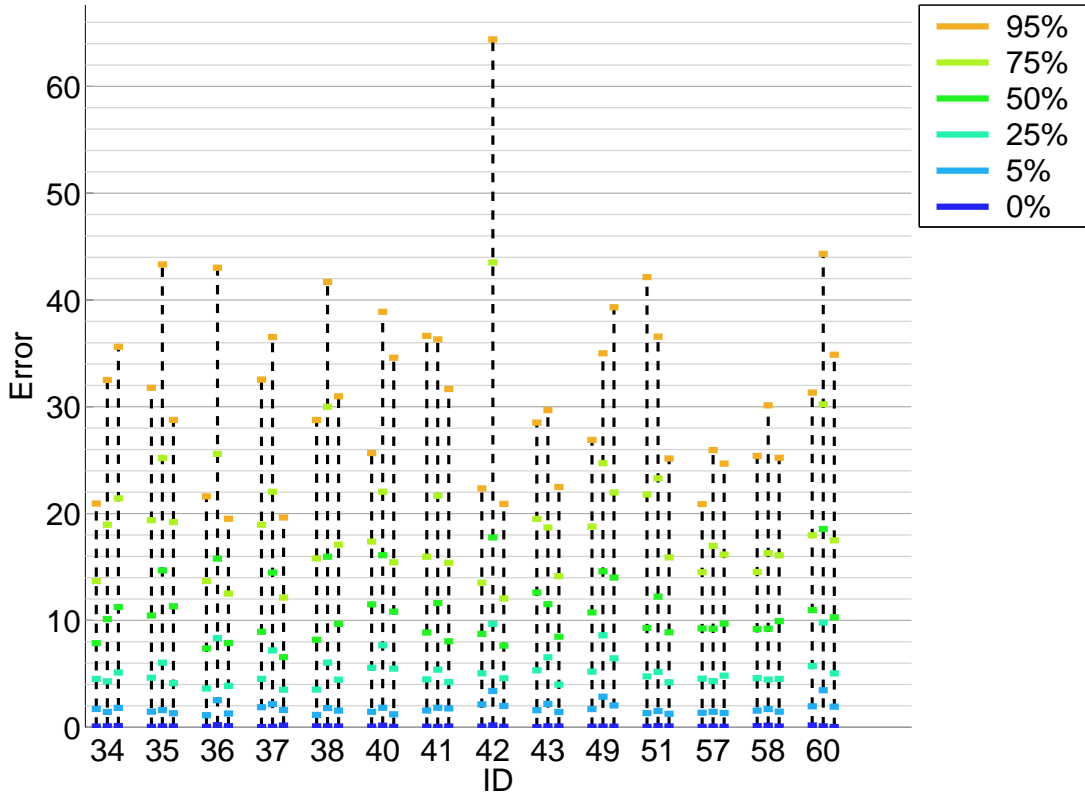
a particle loss test “background likelihood” distribution and a particle loss test threshold c). It was necessary to do this because the components that we changed in each test changed the way in which the particle filter used image information, and so parameters that were optimal in one test were not necessarily optimal in another test on the same video.

The results that we have plotted are based on ground truth estimates that we calculated by fitting deformation fields to manually-tracked landmarks (we will discuss how we did this in detail in chapter 5). For each particle in each time step, we calculated 5 Euclidean distances: the 4 distances between the particle’s vertices and the ground truth positions of the corresponding vertices of the patch we were tracking (under the affine transformations that best fit the deformation fields), and the distance between the particle’s centre and the ground truth position of the corresponding patch’s centre. We then associated the errors for each particle with the particle’s weight, sorted the errors, and calculated percentiles of the resulting weighted error distribution over all frames for each patch that we tracked. These distributions give us an indication of the particle filter’s degree of uncertainty for each patch under each test. We have plotted these percentiles for each test and each video in figure 4.9. Note that the values of all parameters of the particle filter that we specified manually are summarised in tables 3.1 and 6.1 on pages 72 and 211 respectively, with the exception of the components we changed for each comparative test, and of the outlier cutoff p for the \mathcal{E} dissimilarity function that we used in the evaluation of the patch likelihood functions L_t . In our results at the end of the previous chapter we used $p = 0.75$, but we used $p = 0.99$ for the results here, as this seemed to lead to better performance.

Comparing the leftmost columns of percentiles for each patch in the 230108.FO1 video results in figure 4.9a (which represent the error distributions under the methods we have proposed in this chapter) to the central columns (which represent the distributions without lightness normalisation), it is clear that the use of lightness normalisation reduces the errors in the vast majority of cases, sometimes by several pixels (as, for example, in the case of patches 58, 59, 69 and 70). Lightness normalisation only led to much worse results with some of the percentiles of 2 of the 14 patches we tracked: the 50th, 75th and 95th percentiles of patch 71; and the 25th, 50th, 75th, and 95th percentiles of patch 78. It is not entirely clear why this happened, but it



(a) 230108.FO1



(b) 180608.FO1

Figure 4.9: These figures show percentiles of the weighted error distributions that resulted from tracking 14 patches with our proposed method (leftmost column of each group), from tracking the same patches without lightness normalisation (middle columns), and from tracking the patches using the weighted sum-of-squared differences rather than \mathcal{E} (rightmost columns). The IDs identify each of the patches that we tracked in each video. See figure 6.8 on page 212 to see what the patches looked like.

may have been a result of the lightness normalisation process inadvertently destroying some useful textural information about these patches.

Lightness normalisation led to even greater tracking performance improvements in the 180608.HS video, as shown in figure 4.9b. This is to be expected, as there were much greater illumination variations in this video than in the 230108.FO1 video, as mentioned earlier. E.g. compare the 50th, 75th and 95th percentiles of patches 34, 35, 36, 38, 40, 42, 49 and 60 with and without lightness normalisation. The only percentile for which lightness normalisation led to much worse results was the 95th percentile of patch 51.

The use of \mathcal{E} led to less remarkable improvements over our tests using the weighted sum-of-squared differences (the results of which are shown in the rightmost columns of the patch groups in figure 4.9). \mathcal{E} did bring substantial improvements to patches 58, 59, 69 and 70 in 230108.FO1 (the 95th percentiles of which were at least 10 pixels lower with \mathcal{E} . Some of the other percentiles were also much lower under \mathcal{E}), and patches 34 and 49 in 180608.HS (the 95th percentiles of which were at least 12 pixels lower under \mathcal{E}). However, the 95th percentile of patch 60 in 230108.FO1 was 9 pixels higher under \mathcal{E} , and the 75th and 95th percentiles of patches 37, 43 and 51 in 180608.HS were at least 6 pixels higher under \mathcal{E} . For both videos, the differences between many of the remaining percentiles were smaller, and it is not clear if these differences are just due to inaccuracies in the assumptions we make when calculating maximum likelihood estimates of the particle filter parameters, or if they indicate significant differences between \mathcal{E} and the sum-of-squared differences approach.

4.6 Conclusion

In this chapter, we have described the methods that we used to evaluate the patch likelihood function L_t and the local likelihood function ℓ_t . We have shown how significant changes in the appearance of a patch can occur as a result of non-deformational phenomena, such as illumination changes and specular highlights, and we have proposed methods for reducing the extent to which they influence the likelihood functions, such as masking out specular highlights,

and using a “lightness normalisation” method to remove the effects of illumination on the lightness channel with parametric surfaces.

Our comparative results suggest that our lightness normalisation method can lead to substantial improvements in tracking performance, particularly in videos with large illumination variations. The image region comparison function \mathcal{E} that we use to attenuate the influence of unmasked specular highlights and non-affine deformation components has shown some promise, but further tests are required before we can ascertain the situations in which it is most helpful.

Issues to be Addressed in the Remainder of this Thesis

We have now set out the approaches we have taken in designing the main components of our particle filter, but there are two problems related to its performance that must be addressed before reliable, long-term tracking results can be obtained. The rest of this thesis will focus on describing the solutions we have developed to some of these problems, and analysing the performance of these solutions.

The first problem is that the particle filter’s output is heavily dependent on the values chosen for the hyperparameters of the likelihood functions. Setting the standard deviation control parameter, γ , too low or the kurtosis control parameter, α , too high may prevent the importance sampler and patch likelihood evaluator L_t from assigning sufficient probability to the correct state, causing the particle filter to drift. On the other hand, setting γ too high or α too low will cause the importance sampler and patch likelihood evaluator to be insufficiently discriminative, creating posterior distributions with high uncertainty, which will again cause the particle filter to drift. We would ideally like to have an unsupervised method that selects these values based on information from the images.

The second problem is to do with the fact that particles will inevitably drift from time to time. In the best case, such particles will end up with negligible weight, and hence contribute little information about the posterior, in which case updating them will be a waste of computational resources. In the worst case, lots of particles may become lost, impoverishing the particle filter’s representation of the posterior to the point where estimated posterior expectations become

unreliable.

We will address these issues by considering the following questions:

1. How should the kurtosis and inverse decay rate parameters of the likelihood functions be chosen?
2. How can the particle filter determine when a particle is lost?
3. How might the particle filter restore a particle that it thinks is lost?

If a patch is distinguishable from its surroundings, the second of these questions may be dealt with by performing a hypothesis test in which we compare the likelihood of a particle's appearance given its hypothesised state to the likelihood of its appearance given the hypothesis that it has drifted away. We can only perform such tests reliably if we have a good parameterisation of both of these likelihood functions. Thus, the first two questions are closely related, and any solution to them will have to exploit information about the expected changes in the appearance of a patch and its surroundings. The lightness normalisation method we have discussed in this chapter simplifies matters by removing the need to consider appearance changes caused by changing illumination conditions, but we must still consider the effects of myocardial deformations on patch appearance.

So our aims over the next two chapters are as follows:

1. Chapter 5 – to develop statistical models of the sequences of deformations that patches are likely to undergo.
2. Chapter 6 – to use these models to estimate the parameters of the likelihood functions, to develop a hypothesis test for inferring whether or not particles are lost, and to discuss possible ways of restoring lost particles.

The approach we shall take to estimating the likelihood parameters will involve simulating the deformational appearance changes of the patches, and so we will only consider generative statistical deformation models in the next chapter.

Chapter 5

Deformation Modelling

5.1 Synopsis

The work we will present in this chapter paves the way for the solutions we have developed to the first two questions posed in the previous section. As explained in section 3.2, we have assumed the patch deformations to be affine so as to improve the well-posedness of the tracking task and to make it computationally tractable, and in section 3.3.2 we approximated the deformations of subregions by uniformly displacing their pixels. But in reality, the deformations of the myocardium are highly non-linear, and the most significant type of uncertainty about the appearance of a region of the myocardium that must be taken into account when parameterising a likelihood function is the uncertainty introduced by these approximations. Developing a realistic statistical model of the myocardium’s non-linear deformations will allow us to simulate the way they change the appearance of a patch and the appearance of its surroundings, providing an approximation of a data set that we can then use to calculate an initial parameterisation of the likelihood functions.

The deformation models we built were all calculated from our observations of intraoperatively-acquired myocardial image sequences. The model construction procedure is a three-stage process, with each stage involving the following:

1. **Dense Deformation Field Estimation** - For each image sequence, we choose a reference frame t_r , and calculate “smooth” and invertible grids representing the mapping of each point within a region of interest from frame t_r to each successive frame over a time period long enough to cover one cardiac cycle.
2. **Deformation Mode Modelling** - We generate a set of subgrids placed at randomly-selected locations within the domain of the deformation fields, and use the fields to deform the subgrids over the whole image sequence. We remove the affine component of the subgrid deformations, and calculate principal deformation modes with which the dimensionality of the subgrids can be reduced.
3. **Deformation Trajectory Mode Modelling** - We use the reduced-dimensionality deformation model to calculate sequences of parameters from which the subgrid sequences can be reconstructed. Then we calculate principal modes for these parameter sequences, and once again reduce their dimensionality.

In the following sections, we will discuss our implementation of each of these processes.

5.2 Dense Deformation Field Estimation

The first stage of the model construction procedure involves constructing dense deformation fields from which the motion of any point within a region of interest can be calculated. More specifically, given a reference frame t_r and a region of interest \mathcal{R} , we wish to construct vector fields

$$\boldsymbol{\nu}_t : \mathbb{R}^2 \rightarrow \mathbb{R}^2 \tag{5.1}$$

such that $\boldsymbol{\nu}_t(\mathbf{p})$ denotes the position at time t of the point that was at position $\mathbf{p} \in \mathcal{R}$ at time t_r . It is vital for the $\boldsymbol{\nu}_t$ to be invertible, as we will need to explicitly invert them later. The $\boldsymbol{\nu}_t$ should also be reasonably smooth, reflecting the smoothness of the observed deformations. So in short, we may say that the $\boldsymbol{\nu}_t$ should be (approximations of) diffeomorphisms*. For

*A diffeomorphism is a differentiable map with a differentiable inverse. We will not need to calculate derivatives of our deformation fields however, so we do not need them to be differentiable *everywhere*.

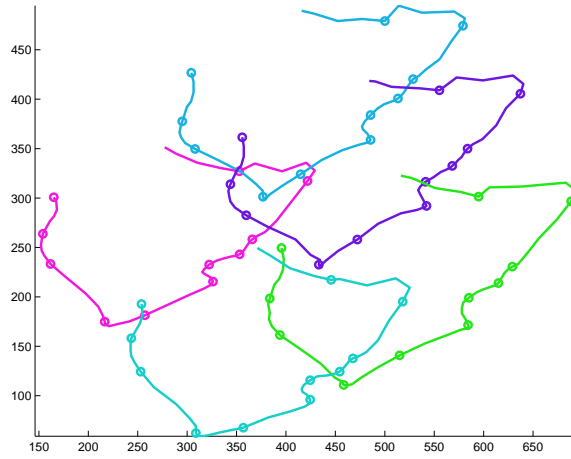


Figure 5.1: An example of the trajectories followed by 5 manually-tracked points over a cardiac cycle, with units in pixels. The circular markers denote the locations of the points at 5-frame intervals. Each trajectory begins at the top-left marker. The range of displacements of the points, relative to their initial positions, is up to 300 pixels horizontally and 200 pixels vertically. The apparent lack of periodicity is due to the effects of respiratory motion.

notational convenience, we will sometimes refer to the corresponding displacement fields

$$\Delta \nu_t(\mathbf{p}) \triangleq \nu_t(\mathbf{p}) - \mathbf{p} \ , \quad (5.2)$$

which give the displacement of point $\mathbf{p} \in \mathcal{R}$ from time t_r to time t .

As shown in figure 5.1, points on the surface of the myocardium undergo very large displacements from their initial position over the course of a cardiac cycle. Most optical flow and image registration algorithms would struggle to calculate such large displacements accurately. The simplest solution to this one might try is to use optical flow to calculate displacement fields $\Delta \hat{\nu}_t$ between consecutive frames $t - 1$ and t , and then define ν_t by composing the $\hat{\nu}_t$:

$$\nu_t = \hat{\nu}_t \circ \dots \circ \hat{\nu}_{t_r+1} \ . \quad (5.3)$$

This may work well when the amount of interframe motion is small. But in the image sequences we have been working with, the interframe motion can be as large as 30-40 pixels. Errors are bound to occur with such large displacements, especially in the large homogeneous regions (shown, for example, in figure 5.3), and over time these errors will accumulate, leading to poor displacement estimates.

More reliable initial estimates of the solution can be obtained by manually tracking as dense a set of landmarks as possible over a cardiac cycle, taking \mathcal{R} to be their convex hull (which can be determined from their Delaunay triangulation), and then estimating the motion at all other points in \mathcal{R} by one of the many interpolation/approximation schemes published in the literature. Any such initial solution estimate $\Delta\boldsymbol{\nu}_{t,0}$ can potentially be refined either by using it as the initial estimate in an optical flow algorithm between images Z_{t_r} and Z_t , or by using $\Delta\boldsymbol{\nu}_{t,0}$ to transform Z_t into an image Z'_t that is more similar to Z_{t_r} by backprojecting Z_t 's pixels:

$$Z'_t(\mathbf{p}) = Z_t(\boldsymbol{\nu}_{t,0}(\mathbf{p})) \quad , \quad (5.4)$$

then calculating the optical flow $\Delta\boldsymbol{\nu}'$ between Z_{t_r} and Z' , and finally defining $\Delta\boldsymbol{\nu}_t$ as an additive update of $\Delta\boldsymbol{\nu}_{t,0}$:

$$\Delta\boldsymbol{\nu}_t(\mathbf{p}) = \Delta\boldsymbol{\nu}_{t,0}(\mathbf{p}) + \Delta\boldsymbol{\nu}'(\mathbf{p}) \quad . \quad (5.5)$$

So, given manually-defined sets of landmarks Q_t such that every point $\mathbf{q}_{t,i} \in Q_t$ is known to correspond to a point $\mathbf{q}_{t',i} \in Q_{t'}$ for all times t and t' at which the landmarks are defined, we want each $\boldsymbol{\nu}_t$ to minimise $\varepsilon^2[\boldsymbol{\nu}_t]$, the sum of squared differences between $\boldsymbol{\nu}_t(Q_{t_r})$ and Q_t :

$$\varepsilon^2[\boldsymbol{\nu}_t] \triangleq \varepsilon_x^2[\boldsymbol{\nu}_{t,x}] + \varepsilon_y^2[\boldsymbol{\nu}_{t,y}] \quad , \quad (5.6)$$

where:

$$\varepsilon_x^2[\boldsymbol{\nu}_{t,x}] \triangleq \sum_{i=1}^{|Q_t|} (\boldsymbol{\nu}_{t,x}(\mathbf{q}_{t_r,i}) - \mathbf{q}_{t,i,x})^2 \quad , \quad \varepsilon_y^2[\boldsymbol{\nu}_{t,y}] \triangleq \sum_{i=1}^{|Q_t|} (\boldsymbol{\nu}_{t,y}(\mathbf{q}_{t_r,i}) - \mathbf{q}_{t,i,y})^2 \quad . \quad (5.7)$$

Clearly, if there is at least one diffeomorphism that interpolates the landmarks, there must be uncountably many (since small changes can be made to $\boldsymbol{\nu}_t$ anywhere except for at the landmarks). Furthermore, there are even more diffeomorphisms that approximate the landmark trajectories to within any given error bound. Thus, to find a unique solution, we must introduce additional constraints.

In [38], large classes of regularising constraints are presented under which unique solutions

to this approximation problem can be obtained. A regulariser will typically be designed to maximise the smoothness of an approximating function, so that any differences in the solutions induced by different regularisers will be the result of differences in the way the regularisers measure smoothness. As ε^2 contains no terms that depend on the fitting error in both the x and y directions, regularising $\boldsymbol{\nu}_t$'s output dimensions independently reduces the approximation problem to that of finding a scalar field for each dimension.

Taking the idea of a function's roughness being a measure of the extent to which it oscillates, the authors of [38] show that a functional $\varsigma[\cdot]$ that measures the roughness of a scalar field f can be defined by measuring the amplitude of the high-frequency components of f 's Fourier transform; i.e., if \tilde{f} is the Fourier transform of f and $\frac{1}{\tilde{G}}$ is a high-pass filter, then

$$\varsigma[f] = \int \frac{|\tilde{f}(\mathbf{s})|^2}{\tilde{G}(\mathbf{s})} d\mathbf{s} , \quad (5.8)$$

where the integration is performed over the frequency domain. The problem of defining f is thus reduced to that of selecting a high-pass filter $\frac{1}{\tilde{G}}$ and a parameter λ that controls the trade-off between minimising ε^2 and minimising ς :

$$f = \arg \min_{f'} \varepsilon^2[f'] + \lambda \varsigma[f'] . \quad (5.9)$$

A popular approximation method (used in many related areas of computer vision, such as interpolating sparse reconstructions of surfaces [44], registration for shape matching [10] and parameterising deformations for 3D myocardial motion tracking [87]) is the Thin-Plate Spline, which filters out low-frequency components with

$$\frac{1}{\tilde{G}(\mathbf{s})} = \|\mathbf{s}\|^4 . \quad (5.10)$$

Filters like this that act isotropically in the Fourier domain (and hence smooth isotropically in the original space) are known as Radial Basis Functions. As shown in [38] (and more formally in [114]), using Radial Basis Function filters like this leads to solutions of eq. (5.9) that just involve solving a linear system of equations. In particular, Thin-Plate Splines give the following

solution for the x ordinates of $\boldsymbol{\nu}_t$ (the uniqueness of this solution was proven in [30]):

$$\boldsymbol{\nu}_{t,x}(\mathbf{p}) = \sum_{i=1}^{|Q_t|} \mathbf{a}_i G(\mathbf{p} - \mathbf{q}_{t_r,i}) + \mathbf{b}^T \mathbf{p} + c , \quad (5.11)$$

where $G(\mathbf{x}) \triangleq \|\mathbf{x}\|^2 \ln \|\mathbf{x}\|$ is the Fourier transform of \tilde{G} , and the $|Q_t|$ -vector \mathbf{a} , 2-vector \mathbf{b} and scalar c are coefficients given by the solution of

$$\left(\begin{array}{c|c} \mathbf{G} + \lambda \mathbf{I} & \mathbf{Q} \\ \hline \mathbf{Q}^T & \mathbf{0} \end{array} \right) \left(\begin{array}{c} \mathbf{a} \\ \mathbf{b} \\ c \end{array} \right) = \left(\begin{array}{c} \mathbf{q}_{t,1,x} \\ \vdots \\ \mathbf{q}_{t,|Q_t|,x} \\ \mathbf{0} \end{array} \right) , \quad (5.12)$$

where \mathbf{G} is the $|Q_t| \times |Q_t|$ matrix $\mathbf{G}_{i,j} \triangleq G(\mathbf{q}_{t_r,i} - \mathbf{q}_{t_r,j})$, and \mathbf{Q} is the $|Q_t| \times 3$ matrix with i^{th} row given by $\mathbf{Q}_{i,:} \triangleq (\mathbf{q}_{t_r,x}, \mathbf{q}_{t_r,y}, 1)$. This can be solved easily using standard linear algebra techniques, such as the LU-decomposition (e.g. see [40]). $\boldsymbol{\nu}_{t,y}$ is defined by solving a similar linear system in which the vector on the right-hand side contains the y ordinates of the \mathbf{q}_t terms.

The Thin-Plate Spline regulariser can be shown to satisfy the following equivalence (see [114]):

$$\varsigma[f] = \int |\tilde{f}(\mathbf{s})|^2 \|\mathbf{s}\|^4 d\mathbf{s} \equiv \int \|\mathbf{H}_f(\mathbf{p})\|_F^2 d\mathbf{p} , \quad (5.13)$$

where $\mathbf{H}_f(\mathbf{p})$ is the Hessian matrix of f at point \mathbf{p} and the norm is the Frobenius norm. From this alternative expression of the regulariser, it becomes clear that the regulariser is minimised when the scalar field f is linear in its argument. This means that $\boldsymbol{\nu}_t$ minimises the regularisers in each output dimension when it is an affine function. In fact, the solution given by eq. (5.11) tends to an affine vector field as $\lambda \rightarrow \infty$. At the other extreme, when $\lambda = 0$, the solution interpolates the landmarks exactly (this follows from the construction of the linear system in eq. (5.12)).

In the ideal case, if the $\boldsymbol{\nu}_t$ could be calculated perfectly, no motion would be visible in the backprojected images \mathbf{Z}'_t (although they may still appear to change with time due to: drifting

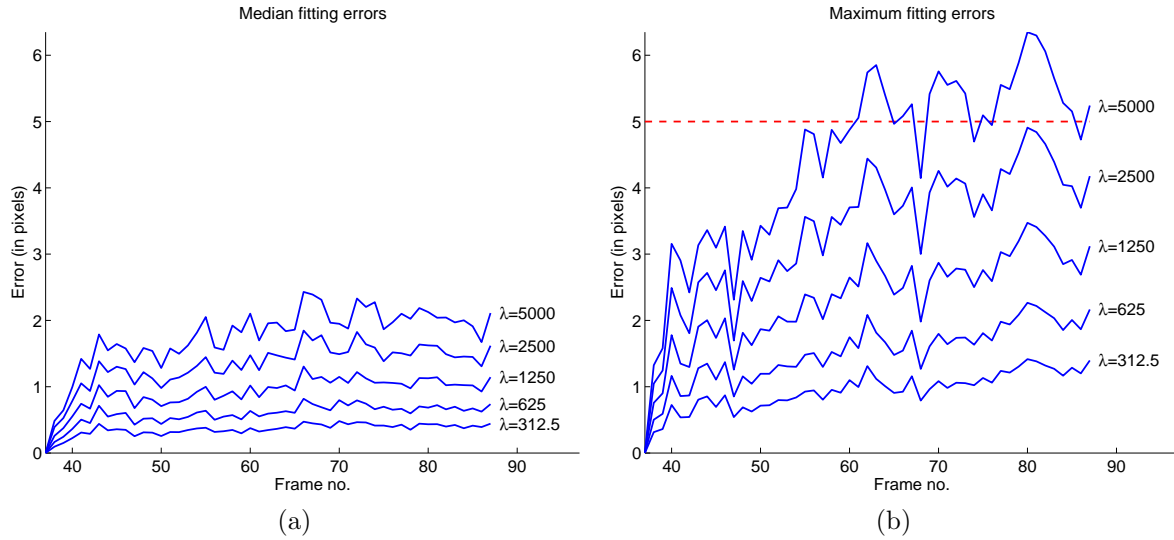


Figure 5.2: (a) This figure shows the median Euclidean error between the true landmark positions and the Thin-Plate Spline estimates of their positions in each frame for various values of λ . (b) This figure shows the maximum Euclidean fitting errors. In this case, the largest value of λ for which the errors are ≤ 5 pixels is $\lambda \approx 2500$.

specular highlights; changes of texture as the myocardium stretches and contracts; illumination changes – we did not make use of lightness normalisation in this chapter, as we calculated point correspondences manually). For low values of the regularisation parameter λ , the \mathbf{Z}'_t s tended to undergo unnatural-looking oscillatory warps, suggesting that $\boldsymbol{\nu}_t$ was overfitting the data. We found that the smoothness brought about by increasing λ seemed to decrease the magnitude of these warps, so we chose the value of λ by manually searching for the largest value that led to a landmark fitting error of no more than about 5–7 pixels*. Unfortunately, it would be difficult to quantify the extent to which increasing λ decreased the warping, as the warping is only noticeable in the regions of \mathbf{Z}'_t that we are unable to reliably track (such as the homogeneous regions), and it is not noticeable in the $\boldsymbol{\nu}_t$ at all. Figures 5.2 and 5.3 show typical examples of our results.

After calculating the Thin-Plate Spline coefficients, we evaluated the Thin-Plate Splines over a regular grid so that we could use a grid-based approximation of the deformation fields in the

*We believe 5 pixels to be a reasonable upper bound on the accuracy with which we can manually track landmarks, and we are willing to accept motion estimation errors of up to 10–12 pixels. If this seems like an overly generous error allowance, the landmark fitting error distributions shown in figure 5.10 on page 162 suggest that the actual motion estimation errors that occurred at the landmarks during our tests were less than 10–12 pixels in the vast majority of cases.

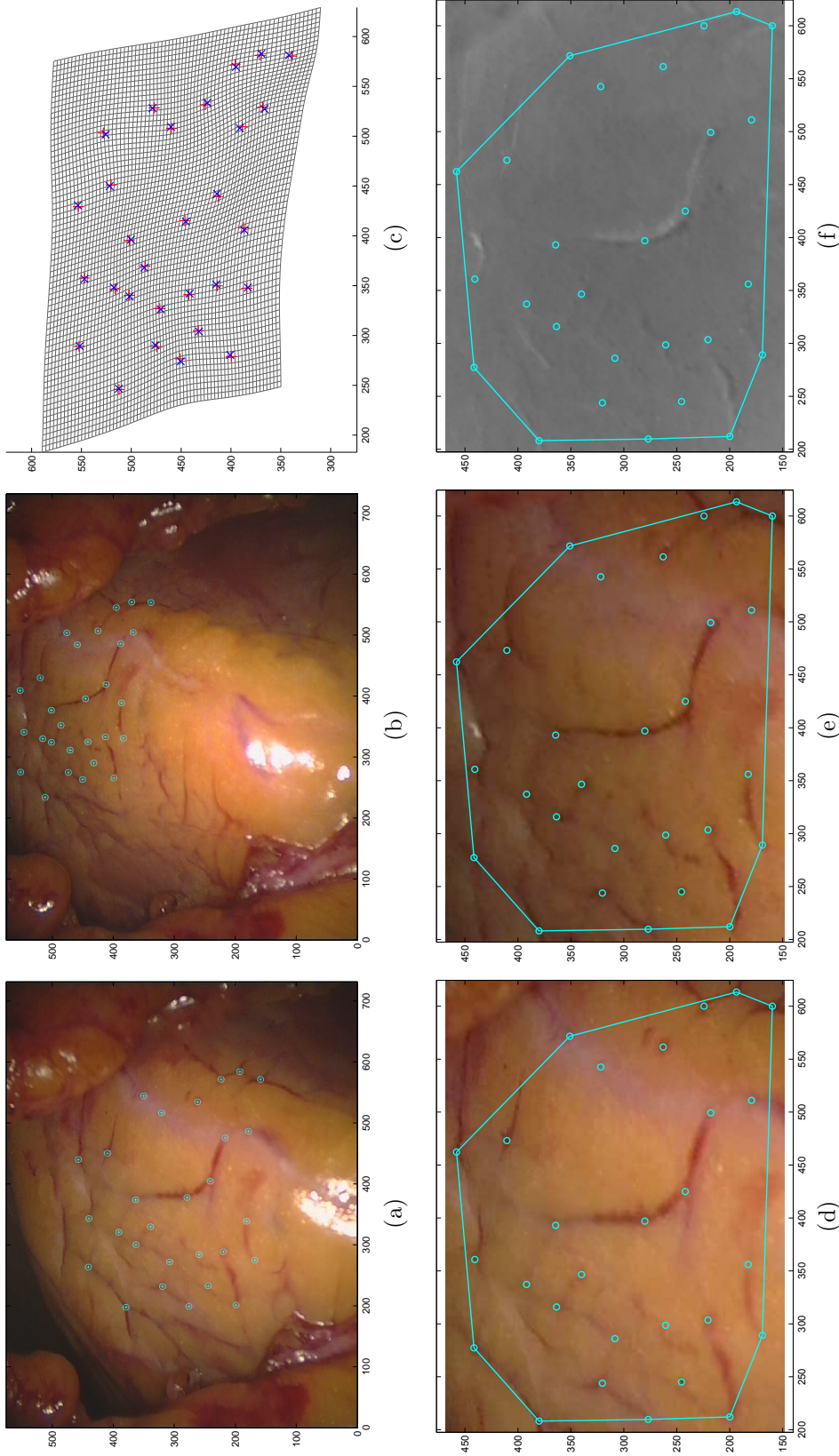


Figure 5.3: (a) The reference image from the 180608.HS sequence with manually-placed landmarks, with units in pixels. Lightness normalisation has not been applied. (b) The image from the sequence that appeared to have the greatest amount of residual motion after estimating the deformation and backprojecting. (c) The estimated deformation, with the landmarks from image (b) shown. The red pluses show the true landmark positions, and the blue crosses show the deformation field's estimates of their positions (d) Close up of the reference image. The indicated convex hull is the region of interest. (e) The result of backprojecting image (b). (f) The difference between the reference image and the backprojected image. Note that difference images do not highlight deformation estimation errors in the homogeneous regions.

remaining work in this chapter. Given a deformation field $\boldsymbol{\nu}_t(\mathbf{p})$, a grid-based approximation $\tilde{\boldsymbol{\nu}}_t(\mathbf{p})$ defined over a regular grid $G \subset \{\mathbf{o}_x + i\delta_g : i \in \mathbb{N}\} \times \{\mathbf{o}_y + i\delta_g : i \in \mathbb{N}\}$ (for some offset \mathbf{o}) returns $\boldsymbol{\nu}_t$'s value at the grid vertices, and approximates it in-between by interpolation. We used bilinear interpolation:

$$\tilde{\boldsymbol{\nu}}_t(\mathbf{p}) = \begin{cases} \boldsymbol{\nu}_t(\mathbf{p}) & , \mathbf{p} \in G \\ (1 - \alpha) \{(1 - \beta)\boldsymbol{\nu}_{i,j} + \beta\boldsymbol{\nu}_{i+1,j}\} + \alpha \{(1 - \beta)\boldsymbol{\nu}_{i,j+1} + \beta\boldsymbol{\nu}_{i+1,j+1}\} & , \text{otherwise,} \end{cases} \quad (5.14)$$

where $\frac{\mathbf{p}_x - \mathbf{o}_x}{\delta_g} \in [i, i + 1)$, $\frac{\mathbf{p}_y - \mathbf{o}_y}{\delta_g} \in [j, j + 1)$ and

$$\boldsymbol{\nu}_{a,b} = \boldsymbol{\nu}_t(\mathbf{o} + \delta_g(a, b)^T) \quad , \quad \alpha = \frac{\mathbf{p}_y - \mathbf{o}_y - j\delta_g}{\delta_g} \quad , \quad \beta = \frac{\mathbf{p}_x - \mathbf{o}_x - i\delta_g}{\delta_g} \quad . \quad (5.15)$$

The grid vertex spacing δ_g can be as small as desired, but it need not be smaller than the largest size at which bilinear interpolation gives an adequate approximation of the Thin-Plate Spline deformations.

The advantages of grid-based representations over Thin-Plate Splines are that: bilinearly-interpolated grid-based deformations can be inverted exactly very efficiently (we will explain how in section 5.3.1), whereas inverting Thin-Plate Splines would require a more difficult, and less efficient, iterative method; the sum of two bilinearly-interpolated grid-based vector fields can be calculated exactly when they are defined on the same grid by simply adding the vectors at each grid vertex (the grids would have to be resampled if they differed), and so they are well-suited to the refinement scheme in eq. (5.5).

In our tests, we found that calculating the Thin-Plate Spline deformation fields between frame t_r and each t directly and then replacing them with a grid-based representation always created invertible transformations. There is no guarantee that this will always be true however. When the result is non-invertible, the frame-by-frame compositional method of eq. (5.3) may be a viable alternative. If that still does not work, the trajectories of the landmarks can be upsampled first, so that the compositional approach can be applied with arbitrarily small displacements between time steps.

5.3 Deformation Mode Modelling

After constructing the deformation fields, the next stage is to develop models of the principal modes of deformation that we can expect the patches we want to track and their immediate surroundings to undergo. As our ultimate goal is to construct a statistical model from which we can simulate myocardial surface deformations, it is beneficial to adopt Occam's razor by searching for the lowest-dimensional representation of the model that explains a sufficient amount of the observable deformation data. The benefits of this are twofold. Firstly, by constructing models that can explain the most significant parts of the observed deformations whilst deviating from them by no more than a tolerable amount, we may be able to filter out some anomalies in the data brought about by errors in the deformation field calculation process. Secondly, minimising the dimensionality of the representation minimises the number of samples that are needed to parameterise it.

As we already have a parameterisation of the affine components of the patch deformations (eq. (3.1)), we neither need nor want our deformation mode models to contain any affine components. So we base the models on an analysis of the residual effects of the deformation fields on a canonical grid \mathbf{g}^* , after removing the affine components of the deformations that they induce on it (the affine components are reintroduced in the final stage, in which we model deformation trajectory modes). The model also needs to incorporate the uncertainties brought about by state information that will be unknown at simulation time, such as the phases of the cardiac cycle that the heart will be in in the frames that we will want to begin the simulations from, and the orientations and positions of the regions we will be simulating deformations of in these frames.

Let T be the length of the cardiac cycle that a deformation field sequence is defined over, so that the fields are defined over frames $T_s = [t_r, \dots, t_r + T - 1]$. We construct a deformation

*We will defer specifying the forms of \mathbf{g} that we used until section 5.3.3. Until then, \mathbf{g} should just be considered to be a fixed configuration of 2D points.

mode model from a set G_S of data sample tuples

$$(\mathbf{o}, \mathbf{f}, g'_{T_s}, h_{T_s}, g_{T_s}) , \quad (5.16)$$

where:

$\mathbf{o} \in \{0, \dots, T-1\}$ is a cardiac phase offset random variable that we use to model the uncertainty concerning the phases that the simulations may start from;

\mathbf{f} is a randomly-selected similarity transformation (i.e. a transformation that only consists of isotropic scaling, translation and rotation) that maps \mathbf{g} into an initial state within frame $t_r + \mathbf{o}$, so as to account for variations in the positions, scales and orientations of the regions we will need to simulate deformations of;

g'_{T_s} is the sequence of deformed instances of $\mathbf{f}(\mathbf{g})$ defined by the deformation fields, under the constraint that $g'_{t_r+\mathbf{o}} = \mathbf{f}(\mathbf{g})$. Since the deformation fields are defined relative to frame t_r , we have to enforce this constraint by defining g'_t in terms of $\boldsymbol{\nu}_{t_r+\mathbf{o}}$'s inverse $\boldsymbol{\nu}_{t_r+\mathbf{o}}^{-1}$:

$$g'_t \triangleq \boldsymbol{\nu}_t \left(\boldsymbol{\nu}_{t_r+\mathbf{o}}^{-1}(\mathbf{f}(\mathbf{g})) \right) ; \quad (5.17)$$

h_{T_s} is a sequence of transformations that affinely registers each g'_t to \mathbf{g} ;

g_{T_s} is the result of registering each g'_t to \mathbf{g} with h_t . It represents the sequence of non-affine deformation components.

Since: the regions we will be simulating deformations of could lie anywhere on the visible part of the myocardium; we have no prior information about what the distance of the endoscope from the myocardium will be or what its orientation about the viewing direction will be; and we may need to begin the simulation at any cardiac phase; we take the most noncommittal approach of sampling the pair (\mathbf{o}, \mathbf{f}) from a distribution that is close to uniform (we will define the sampling procedure precisely in section 5.3.3). Bounds are imposed on this distribution by the constraint that $\mathbf{f}(\mathbf{g})$ must lie within $\boldsymbol{\nu}_{t_r+\mathbf{o}}$'s image, so that we can calculate g'_{T_s} . The process of constructing G_S is summarised in algorithm 5.1.

Algorithm 5.1: Constructing the deformation mode model sample data set G_S .

Result: A set G_S of N data sample tuples.

 $G_S := \emptyset$
for $i := 1$ **to** N **do**
 $(\mathbf{o}, \mathbf{f}) :=$ a (cardiac phase offset, similarity transformation) pair drawn from an approximately uniform distribution bounded by the constraint that $\mathbf{f}(\mathbf{g})$ lies in the image of $\boldsymbol{\nu}_{t_r+\mathbf{o}}$.

 $t := t_r$
for $j := 1$ **to** T **do**
 $g'_t := \boldsymbol{\nu}_t(\boldsymbol{\nu}_{t_r+\mathbf{o}}^{-1}(\mathbf{f}(\mathbf{g})))$
 $h_t :=$ an affine transformation that registers g'_t to \mathbf{g} .

 $g_t := h_t(g'_t)$
 $t := t + 1$
end
 $G_S := G_S \cup \{(\mathbf{o}, \mathbf{f}, g', h, g)\}$
end

It is important to mention that our construction of G_S is unable to adequately take into account perspective changes caused by variations in the angles of inclination of the regions we will be simulating deformations of, since we do not have access to depth information. Knowing the depth of the pixels in the image sequences would allow us to construct the deformation fields and deformation mode models in 3D, which would lead to simulations in which perspective changes emerge naturally. Without depth information, the best we can do is to construct sample sets from multiple deformation field sequences and merge them together, so that perspective changes will be “baked into” the deformation mode and deformation trajectory mode models.

In the next two sections we will explain how to invert the deformation fields and how to carry out the affine registration of two grids. After that, we will discuss the details of the various approaches to deformation mode calculation that we employed.

5.3.1 Deformation Field Inversion

As we use a grid-based representation for the deformation fields, calculating $\boldsymbol{\nu}_t^{-1}(\mathbf{p})$ involves identifying the square S of the grid G that satisfies $\mathbf{p} \in \boldsymbol{\nu}_t(S)$, and then calculating the interpolation coefficients of the point in S that $\boldsymbol{\nu}_t$ maps to \mathbf{p} . S can be determined efficiently

by the use of an inverse lookup table, which only needs to be calculated once for each ν_t .

The lookup table is defined by a finite regular grid G' that covers ν_t 's image such that each square S' of G' is associated with a subset $H(S')$ of G 's grid squares. The grid spacing of G' we use is the mean side length of $\nu_t(G)$'s quadrilaterals. As it is trivial to identify the square $S'(\mathbf{p})$ of G' containing the point \mathbf{p} that we want to apply the inverse transformation to, defining $H(S'(\mathbf{p}))$ such that

$$S \in H(S'(\mathbf{p})) \leftarrow \nu_t(S) \cap S'(\mathbf{p}) \neq \emptyset \quad (5.18)$$

means that we only need to search $H(S'(\mathbf{p}))$ to find the square $S(\mathbf{p})$ of G that contains $\nu_t^{-1}(\mathbf{p})$. We identify the square by verifying that $\nu_t(S(\mathbf{p})) = \square abdc$ contains \mathbf{p} (see figure 5.4b). The containment test is performed on triangles $\triangle adc$ and $\triangle abd$, using the fact that a point \mathbf{q} lies in triangle $\triangle xyz$ if and only if \mathbf{q} is on the same side of line: \overrightarrow{xy} as z , \overrightarrow{xz} as y and \overrightarrow{yz} as x , which in turn can be determined using the fact that \mathbf{q} lies on the same side of a line \overrightarrow{rs} as a point \mathbf{u} if and only if

$$(\overrightarrow{rq} \times \overrightarrow{rs})(\overrightarrow{ru} \times \overrightarrow{rs}) \geq 0 \quad , \quad (5.19)$$

where \times denotes the 2D cross product:

$$\mathbf{v} \times \mathbf{w} \triangleq v_x w_y - v_y w_x \quad . \quad (5.20)$$

Although the optimal way to define which H sets to add each square S of G to would be to rasterise $\nu_t(S)$ with respect to G' , it is much simpler and not much less efficient to just add S to each $H(S')$ for which S' intersects $\nu_t(S)$'s bounding box, as shown in figure 5.4a.

The final step of calculating the interpolation coefficients α and β (as in eq. (5.14)) that define $\nu_t^{-1}(\mathbf{p})$ involves solving a quadratic equation. With reference to figure 5.4b, we use the solution given in [16], where the author points out that the α parameter takes the unique value that makes the points $\mathbf{a} + \alpha \overrightarrow{ac}$ and $\mathbf{b} + \alpha \overrightarrow{bd}$ collinear with \mathbf{p} . This collinearity requirement is equivalent to stating that the sine of the angle between $\mathbf{a} + \alpha \overrightarrow{ac} - \mathbf{p}$ and $\mathbf{b} + \alpha \overrightarrow{bd} - \mathbf{p}$ is zero,

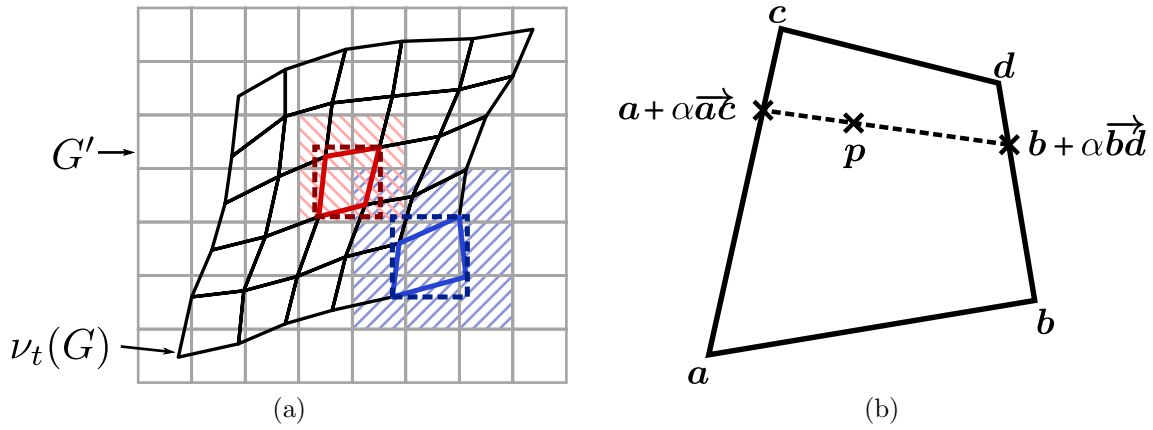


Figure 5.4: (a) The time t_r regular grid G is deformed by ν_t . Each coloured quadrilateral $\nu_t(S)$ in the deformed grid corresponding to a grid square S of G has a bounding box indicated with dashed lines and is added to $H(S')$, for each square S' of G' shaded in a similar colour to $\nu_t(S)$. (b) The bilinear interpolation coefficient α , which is used in the calculation of the position of $\nu_t^{-1}(p)$ in the grid square of G that is mapped to $\square abdc$, makes $a + \alpha \vec{ac}$, p and $b + \alpha \vec{bd}$ collinear.

which implies that their cross product is zero, which in turn implies:

$$(\vec{ac} \times \vec{bd})\alpha^2 + (\vec{bd} \times \vec{ap} - \vec{ac} \times \vec{bp})\alpha + (\vec{ap} \times \vec{bp}) = x\alpha^2 + y\alpha + z = 0. \quad (5.21)$$

As explained in [16], a numerically-stable solution to this quadratic equation that lies in the interval $[0, 1]$ is

$$\alpha = \begin{cases} \frac{-y - \sqrt{y^2 - 4xz}}{2x} & , y > 0 \\ \frac{2z}{-y + \sqrt{y^2 - 4xz}} & , \text{otherwise.} \end{cases} \quad (5.22)$$

For non-degenerate quadrilaterals, $x = 0$ implies that \vec{ac} and \vec{bd} are parallel, which in turn implies that the angle from \vec{bd} to \vec{ap} is ≤ 0 and the angle from \vec{ac} to \vec{bp} is ≥ 0 . At most one of these inequalities can be equal to zero, so $y < 0$ and the second case is executed. Also, $z = 0$ implies that \vec{ap} is collinear with \vec{bp} , but y is non-zero in this case for non-degenerate quadrilaterals. So this solution only fails when $x = y = 0$, which occurs when either $b = d = p$ or $a = c = p$, or when $\square abdc$'s vertices are collinear. Such degeneracies never occur in our application, as they could only be caused by physically impossible deformations.

The β parameter is proportional to the projection of $\mathbf{p} - (\mathbf{a} + \alpha \vec{\mathbf{a}\mathbf{c}})$ onto $\mathbf{b} + \alpha \vec{\mathbf{b}\mathbf{d}} - (\mathbf{a} + \alpha \vec{\mathbf{a}\mathbf{c}})$:

$$\beta = \frac{\|\mathbf{p} - (\mathbf{a} + \alpha \vec{\mathbf{a}\mathbf{c}})\|}{\|\mathbf{b} + \alpha \vec{\mathbf{b}\mathbf{d}} - (\mathbf{a} + \alpha \vec{\mathbf{a}\mathbf{c}})\|} . \quad (5.23)$$

Finally, we have

$$\boldsymbol{\nu}_t^{-1}(\mathbf{p}) = S(\mathbf{p})_a + \alpha(S(\mathbf{p})_c - S(\mathbf{p})_a) + \beta(S(\mathbf{p})_b - S(\mathbf{p})_a) , \quad (5.24)$$

where $S(\mathbf{p})_a$, $S(\mathbf{p})_b$ and $S(\mathbf{p})_c$ are the vertices of $S(\mathbf{p})$ that correspond to \mathbf{a} , \mathbf{b} and \mathbf{c} .

5.3.2 Affine Point Set Registration

The affine components of the transformations that map the grids of the sequence g'_{T_s} to the canonical grid \mathbf{g} can be expressed as the solutions of least squares optimisation problems. In sections 5.3.4 and 5.3.5, we will describe how we construct deformation mode models under two different types of canonical grid, each of which defines the affine component as the minimum of a different type of optimisation problem. We will define and solve these problems now, so as not to break up the discourse when we describe the deformation mode models.

Let \mathbf{Y} and \mathbf{X} be $d \times N$ matrices of N d -dimensional column vectors representing \mathbf{g} and one of the grids of g'_{T_s} respectively*. The first optimisation problem is the unconstrained problem of calculating a $d \times d$ matrix \mathbf{M} that minimises the sum of squared differences between the vectors of \mathbf{MX} and \mathbf{Y} :

$$\mathbf{M} = \arg \min_{\mathbf{M}'} \|\mathbf{MX} - \mathbf{Y}\|_F^2 . \quad (5.25)$$

The second is the constrained problem of minimising eq. (5.25) subject to

$$\mathbf{Ma} = \mathbf{b} , \quad (5.26)$$

for some d -vectors \mathbf{a} and \mathbf{b} . Although, in both cases, we require \mathbf{M} to have strictly-positive

*To account for translation, we will represent 2D affine transformations in homogeneous coordinates, for which $d = 3$.

determinant, it is not necessary to enforce this algebraically, since the deformation fields that the $G_{C,1:t}$ are calculated from contain no reflective components.

It should be noted that the popular Procrustes Analysis method (described, for example, in [43] and [94]) is too restrictive to solve these problems, since it only finds the best-fitting similarity transformations. Also, although our aim is to bring all of the grids in the g'_{T_s} sequences into a common coordinate system, it is unnecessary for us to perform an affine equivalent of the iterative Generalised Procrustes Analysis method developed by Gower in [42], since the g'_{T_s} sequences are all deformations of the pre-defined canonical grid \mathbf{g} .

Unconstrained Affine Registration

The unconstrained problem is equivalent to solving d linear least squares problems, one for each row of \mathbf{M} and \mathbf{Y} :

$$\mathbf{M}_{i,:} = \arg \min_{\mathbf{c}^T} \|(\mathbf{Y}_{i,:})^T - \mathbf{X}^T \mathbf{c}\|^2 . \quad (5.27)$$

The solution to this is well-known, and is given by the Moore-Penrose pseudoinverse $(\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}$ of \mathbf{X}^T (see [78] for the derivation):

$$(\mathbf{M}_{i,:})^T = (\mathbf{X}\mathbf{X}^T)^{-1}\mathbf{X}(\mathbf{Y}_{i,:})^T . \quad (5.28)$$

Stacking these solutions for each row of \mathbf{M} gives:

$$\mathbf{M} = \mathbf{Y}\mathbf{X}^T(\mathbf{X}\mathbf{X}^T)^{-1} . \quad (5.29)$$

Constrained Affine Registration

The constraint $\mathbf{M}\mathbf{a} = \mathbf{b}$ removes d degrees of freedom from \mathbf{M} . Equivalently, considering each row of \mathbf{M} to be a d -vector, the constraint tells us that $\mathbf{M}_{i,:}$ lies on the hyperplane with normal $\frac{\mathbf{a}}{\|\mathbf{a}\|}$ that lies at distance $\frac{\mathbf{b}_i}{\|\mathbf{a}\|}$ from the origin. By this latter interpretation, each $\mathbf{M}_{i,:}$ can be parameterised by an unconstrained $(d-1)$ -dimensional column vector $\mathbf{W}_{:,i}$ that defines a point

on the hyperplane. The problem is thus reduced to that of finding the $(d-1) \times d$ matrix \mathbf{W} that solves

$$\mathbf{W} = \arg \min_{\mathbf{W}'} \|(\mathbf{W}'^T \mathbf{B}^T + \mathbf{Q}^T) \mathbf{X} - \mathbf{Y}\|_F^2, \quad (5.30)$$

after which \mathbf{M} is given by

$$\mathbf{M} = \mathbf{W}^T \mathbf{B}^T + \mathbf{Q}^T, \quad (5.31)$$

where \mathbf{B}^T is a $(d-1) \times d$ matrix of rank $d-1$ that satisfies

$$\mathbf{B}^T \mathbf{a} = \mathbf{0}, \quad (5.32)$$

and \mathbf{Q}^T is an arbitrary $d \times d$ matrix that satisfies

$$\mathbf{Q}^T \mathbf{a} = \mathbf{b}. \quad (5.33)$$

Given \mathbf{B} and \mathbf{Q} , eq. (5.30) is another linear least squares problem, which is solved this time by the pseudoinverse of $(\mathbf{B}^T \mathbf{X})^T$:

$$\begin{aligned} \mathbf{W} &= \arg \min_{\mathbf{W}'} \|\mathbf{W}'^T \mathbf{B}^T \mathbf{X} - (\mathbf{Y} - \mathbf{Q}^T \mathbf{X})\|_F^2 \\ &= ((\mathbf{Y} - \mathbf{Q}^T \mathbf{X})(\mathbf{B}^T \mathbf{X})^T (\mathbf{B}^T \mathbf{X} (\mathbf{B}^T \mathbf{X})^T)^{-1})^T, \text{ by analogy with eq. (5.29)} \\ &= (\mathbf{B}^T \mathbf{X} \mathbf{X}^T \mathbf{B})^{-1} \mathbf{B}^T \mathbf{X} (\mathbf{Y}^T - \mathbf{X}^T \mathbf{Q}), \end{aligned} \quad (5.34)$$

The rows of \mathbf{B}^T define a basis (up to a displacement) for the subspaces occupied by the d parallel hyperplanes that the rows of \mathbf{M} lie on (or equivalently, the columns of \mathbf{B} form a basis for \mathbf{a}^T 's null space), and each row of \mathbf{Q}^T is a point on the corresponding hyperplane. The choice of \mathbf{B} and \mathbf{Q} that satisfy eqs. (5.32) and (5.33) is arbitrary. We define \mathbf{Q}^T as

$$\mathbf{Q}^T = \left(\overbrace{\mathbf{0} \ \cdots \ \mathbf{0}}^{\times(i-1)} \ \frac{\mathbf{b}}{\mathbf{a}_i} \ \mathbf{0} \ \cdots \right), \quad (5.35)$$

where i is an arbitrary index of \mathbf{a} such that $\mathbf{a}_i \neq 0$. \mathbf{B}^T can be defined by the Singular-Value Decomposition of \mathbf{a} (which can be calculated using the methods described in [40], and is implemented in many numerical libraries):

$$\mathbf{a} = \mathbf{U}\boldsymbol{\sigma}v, \quad (5.36)$$

where \mathbf{U} is an orthogonal $d \times d$ matrix, $\boldsymbol{\sigma}$ is a d -dimensional column vector such that $\sigma_{2:d} = 0$, and v is either 1 or -1. This implies that

$$\mathbf{a} = \mathbf{U}_{:,1}\sigma_1v. \quad (5.37)$$

As \mathbf{U} is orthogonal, it follows that the columns of $\mathbf{U}_{:,2:d}$ are orthogonal to each other and to \mathbf{a} . So we can take $\mathbf{U}_{:,2:d}$ as our rank $d - 1$ subspace basis \mathbf{B} .

5.3.3 Sampling Phase Offsets and Canonical Grid Similarity Transformations

When constructing a deformation model, a choice has to be made about what representation of the sample data to base the model on. There are many possible choices, and it is not immediately clear what choice would yield the lowest-dimensional model. In our work we have tested two possible choices, which will be described in the next section. First, we will discuss the two types of canonical grid we defined to go with these choices, and we will show how we sampled the similarity transformations that we used to define the sample sets.

The two types of canonical grid we used were: a zero-centred regular grid, \mathbf{g}_s , with $l \times l$ vertices evenly distributed over the unit square; and a zero-centred quasi-circular grid, \mathbf{g}_c , of unit area, defined by r uniformly-spaced concentric rings (each of which is a regular s -sided polygon, except for the innermost ring, which is degenerate) with s uniformly-spaced spokes delineating s similar isosceles-triangle-shaped segments of the outer ring. Typical examples of the grids are given in figure 5.5. We will use \mathbf{g} when discussing things that apply to both grids.

We use a zero-based 2D index (i, j) to refer to the vertices of each grid, where i and j identify the row and column respectively of \mathbf{g}_s 's vertices:

$$\mathbf{g}_s(i, j) \triangleq \frac{1}{l-1} \begin{pmatrix} j - \frac{l-1}{2} \\ i - \frac{l-1}{2} \end{pmatrix}, \quad (5.38)$$

and the ring and segment respectively of \mathbf{g}_c 's vertices:

$$\mathbf{g}_c(i, j) \triangleq \mathbf{r}_{c,s}(1) \frac{i}{(r-1)} \begin{pmatrix} \cos\left(2\pi \frac{j}{s}\right) \\ \sin\left(2\pi \frac{j}{s}\right) \end{pmatrix}. \quad (5.39)$$

The function $\mathbf{r}_{c,s}(a)$ gives the radius of the regular s -sided polygon of area a , and it is used here to give \mathbf{g}_c unit area. It follows from elementary trigonometry that an isosceles triangle with apex angle $\frac{2\pi}{s}$ and side length (i.e. outer ring radius) l has area $l^2 \cos\left(\frac{\pi}{s}\right) \sin\left(\frac{\pi}{s}\right)$. Multiplying by s , making use of the double-angle formula and rearranging gives $\mathbf{r}_{c,s}(a)$:

$$\mathbf{r}_{c,s}(a) \triangleq \sqrt{\frac{2a}{s \sin\left(\frac{2\pi}{s}\right)}}. \quad (5.40)$$

As with the grid-based deformation fields, each vertex (i, j) in \mathbf{g}_s is connected to its four (two or three at the boundary) neighbours $(i-1, j)$, $(i+1, j)$, $(i, j-1)$ and $(i, j+1)$, and similarly, each vertex (i, j) in \mathbf{g}_c 's interior is connected to its four (three at the boundary) neighbours $(i-1, j)$, $(i+1, j)$, $(i, (j-1) \pmod s)$ and $(i, (j+1) \pmod s)$. We model deformations by considering the motion of the vertices, and again, we approximate the motion at all other points within the grids by bilinearly interpolating over the quadrilaterals defined by the grids' edges.

As mentioned earlier, we need to randomly sample phase offsets and similarity transformations (\mathbf{o}, \mathbf{f}) for the canonical grids over a domain Ω that has bounds determined by the constraint that $\mathbf{f}(\mathbf{g})$ must lie within $\mathbf{v}_{t_r+\mathbf{o}}$'s image $\mathcal{S}_{\mathbf{o}}$. We can draw uniform samples from Ω by rejection sampling. As \mathbf{f} is determined by a 2D displacement \mathbf{p} , a grid area \mathbf{a} and a rotation Φ , the simplest form of rejection sampling would involve defining a 5D cuboid $\Omega^* \supset \Omega$, and repeatedly drawing samples from it (sampling from each dimension independently) until we draw one that satisfies the constraint. But we can sample more efficiently by making use of some geometric

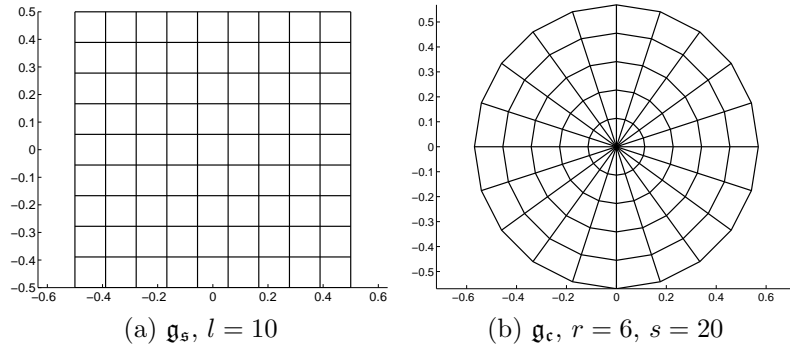


Figure 5.5: (a) The 10×10 square canonical grid that we used in our experiments, with $2 \times 10 \times 10 = 200$ degrees of freedom. (b) A 6 ring, 20 sector quasi-circular canonical grid that we used. The 20 points of the degenerate innermost ring will always be collocated after any transformation, hence the grid has $2 \times (5 \times 20 + 1) = 202$ degrees of freedom. The two grids have equal area and a similar number of degrees of freedom.

constraints.

The first constraint, which holds for both canonical grids, is that for any \mathbf{o} , the displacement \mathbf{p} must lie within $\mathcal{S}_{\mathbf{o}}$. By making use of the Delaunay triangulation of the landmarks that define $\mathcal{S}_{\mathbf{o}}$, we can sample \mathbf{p} within $\mathcal{S}_{\mathbf{o}}$ by selecting a triangle $\triangle \mathbf{wxy}$ with a probability proportional to its area*, and then sampling a point within it. As explained in [117], one way to sample uniformly over $\triangle \mathbf{wxy}$ is to begin by drawing a sample \mathbf{p}' over the parallelogram $\square \mathbf{wxzy}$, where $\mathbf{z} = \mathbf{x} + \overrightarrow{\mathbf{wy}}$ (see figure 5.6a):

$$\mathbf{p}' = \mathbf{w} + U_0 \overrightarrow{\mathbf{wx}} + U_1 \overrightarrow{\mathbf{wy}}, \quad (5.41)$$

for i.i.d. standard uniform variables U_0 and U_1 . Then we set \mathbf{p} to \mathbf{p}' if it lies within $\triangle \mathbf{wxy}$. Otherwise, \mathbf{p}' lies within $\triangle \mathbf{xyz}$, and we map it to a point \mathbf{p} in $\triangle \mathbf{wxy}$ using a linear mapping that maps \mathbf{z} to \mathbf{w} , keeps the line through \mathbf{x} and \mathbf{y} fixed, and that satisfies $\frac{\|\overrightarrow{\mathbf{zp}'}\|}{\|\overrightarrow{\mathbf{zq}}\|} = \frac{\|\overrightarrow{\mathbf{wp}}\|}{\|\overrightarrow{\mathbf{wq}}\|}$,

*Note: when calculating our results at the end of this chapter, we mistakenly selected the triangles with uniform probability instead. In theory, this would have had the effect of biasing the sample set towards the deformations around small triangles. However, our cross-validation results in figure 5.14 on 167 showed that the deformation mode models performed more-or-less as well on images that they were not trained on as they did on images that they were trained on, suggesting that the biasing effect was insignificant.

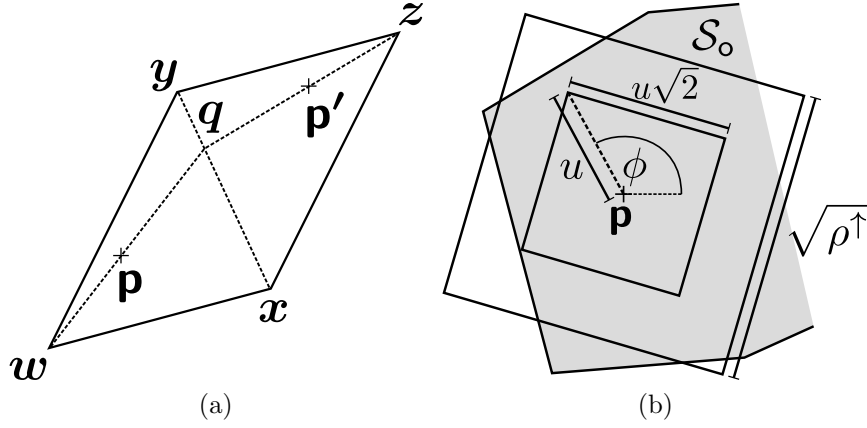


Figure 5.6: (a) A point \mathbf{p}' uniformly sampled within parallelogram $\square wxyz$ can be linearly mapped to a point \mathbf{p} in triangle $\triangle wxy$, giving a uniformly-distributed sample within that triangle. (b) After selecting \mathbf{o} , \mathbf{p} and Φ , the smallest distance u from \mathbf{p} to \mathcal{S}_o 's boundary determines the upper bound on the scale factor \mathbf{a} .

where \mathbf{q} is the intersection of the line through \mathbf{z} and \mathbf{p}' with the line through \mathbf{x} and \mathbf{y} :

$$\begin{aligned}\mathbf{p} &= \mathbf{w} + \frac{\|\overrightarrow{z\mathbf{p}'}\|}{\|\overrightarrow{z\mathbf{q}}\|} \overrightarrow{w\mathbf{q}} \\ \mathbf{q} &= \mathbf{x} + v\overrightarrow{xy},\end{aligned}\tag{5.42}$$

and the scalar v is the parameter such that the line $\mathbf{x} + v\overrightarrow{xy}$ intersects the line through \mathbf{z} and \mathbf{p}' :

$$\mathbf{x} + v\overrightarrow{xy} - \mathbf{z} \propto \overrightarrow{z\mathbf{p}'} \quad \Rightarrow \quad \left(\overrightarrow{p'z} \mid \overrightarrow{xy} \right) \begin{pmatrix} \cdot \\ v \end{pmatrix} = \overrightarrow{xz}.\tag{5.43}$$

The second constraint is that \mathbf{g} 's boundary must lie within \mathcal{S}_o . We bound the area \mathbf{a} above at a manually-selected value a^\uparrow that ensures that $\mathbf{f}(\mathbf{g})$ cannot contain \mathcal{S}_o . So the second constraint is satisfied when none of the line segments from \mathbf{p} to $\mathbf{f}(\mathbf{g})$'s boundary vertices cross \mathcal{S}_o 's boundary. For \mathbf{g}_s , after selecting \mathbf{o} and \mathbf{p} , we randomly sample a direction $\mathbf{v} = (\cos(\Phi), \sin(\Phi))^T$ for the vector from \mathbf{p} to one of the boundary vertices by sampling the orientation Φ over $[0, 2\pi)$. Then we take the upper bound on \mathbf{a} to be at most $2u^2$ (the area of a square with diagonal length $2u$), where $u \geq 0$ is the smallest radius such that one of $\mathbf{f}(\mathbf{g}_s)$'s four corners, $\mathbf{p} + u\mathbf{v}$, $\mathbf{p} - u\mathbf{v}$, $\mathbf{p} + u(-\mathbf{v}_y, \mathbf{v}_x)^T$ or $\mathbf{p} - u(-\mathbf{v}_y, \mathbf{v}_x)^T$, intersects \mathcal{S}_o 's boundary. We search for u by calculating the intersections of the lines through \mathbf{p} and each of these corners with the line through \mathbf{a} and \mathbf{b} (given by the solutions of simple linear equations analogous to eq. (5.43)) for each line segment

from \mathbf{a} to \mathbf{b} on \mathcal{S}_o 's boundary. We also bound \mathbf{a} below at a value a^\downarrow that defines the area of the smallest grids that we are interested in modelling the deformations of. So if $2u^2 < a^\downarrow$, we reject \mathbf{p} , \mathbf{o} and Φ and start again. Otherwise, we sample \mathbf{a} over $[a^\downarrow, \min(2u^2, a^\uparrow)]$.

A similar method could be used to sample \mathbf{a} for \mathbf{g}_c , but for large s it is more efficient to work with the smallest circle centred at \mathbf{p} that contains \mathbf{g}_c for any given value of \mathbf{a} . Hence, we take the upper bound on \mathbf{a} to be $\min(\pi u'^2, a^\uparrow)$ and reject the current samples if $\pi u'^2 < a^\downarrow$, where u' is the smallest perpendicular distance from \mathbf{p} to \mathcal{S}_o 's boundary. This only requires n intersection calculations, as opposed to the sn calculations required by the previous method, where n is the number of line segments that make up \mathcal{S}_o 's boundary. Working with circles means that the value of the orientation parameter Φ has no bearing on the issue of whether or not \mathcal{S}_o contains $\mathbf{f}(\mathbf{g}_c)$ (for fixed \mathbf{o} and \mathbf{p}), but it also has the effect of trimming away some parts of Ω near the boundary of \mathcal{S}_o (for each value of \mathbf{o}). However the ($s = 20$)-segment grid we performed our experiments with is so close to being perfectly circular (it occupies about 98% of the area of a circle with the same radius) that the volume of Ω that is lost will generally be very small in comparison to its original volume.

Although rejection sampling like this gives uniformly-distributed (\mathbf{o}, \mathbf{f}) samples, the marginal distributions of \mathbf{a} produced by these methods tend to be heavily skewed towards small \mathbf{a} , since each \mathcal{S}_o can contain more small grids than large grids, and in some cases there were large changes in the area of \mathcal{S}_o over the deformation field sequences (e.g. see the “180608.HS” row in table 5.4). Rather than using uniformly-distributed (\mathbf{o}, \mathbf{f}) samples with these skewed \mathbf{a} marginals, it is more important to us to generate samples that have a uniform marginal distribution over \mathbf{a} , and a conditionally-uniform distribution over \mathbf{o} , \mathbf{p} and Φ given \mathbf{a} . This should make for fairer comparisons of the performance of \mathbf{g}_s -based models and \mathbf{g}_c -based models (since the marginal distributions over \mathbf{a} would probably be different otherwise), and it should make the models attach as much importance to the kinds of deformations that mostly only occur at large scales as they do to those that mostly only occur at smaller scales. So, to make \mathbf{a} 's marginal distribution more uniform, we partition $[a^\downarrow, a^\uparrow]$ into m subsets of equal size, and then sample m groups of $\frac{N}{m}$ grids, taking the bounds of each subset as \mathbf{a} 's sampling range for each group (as in algorithm 5.1, N is the total number of grids to be transformed, and we assume that m

divides N).

Finally, before \mathbf{f} 's parameters can be sampled, we have to choose a value for \mathbf{o} . The only difficulty here is that of determining whether or not we have selected an \mathbf{o} for which any $\mathbf{f}(\mathbf{g})$ lies in $\mathcal{S}_{\mathbf{o}}$. We use a probabilistic argument to infer whether or not this is true. For each (\mathbf{o}, \mathbf{f}) sample we wish to draw, we initialise \mathbf{o} 's domain to the set $X = \{0, \dots, T-1\}$, draw a sample \mathbf{o}' from it and attempt to sample \mathbf{f} 's parameters as described above. If we are unable to sample \mathbf{f} before M rejections are made, we remove \mathbf{o}' from X , and begin again. Now, for any \mathbf{o}' such that $\mathcal{S}_{\mathbf{o}'}$ contains some $\mathbf{f}(\mathbf{g})$ s, the number of rejections that will be made before a contained $\mathbf{f}(\mathbf{g})$ is found follows a geometric distribution. The probability of finding an $\mathbf{f}(\mathbf{g})$ in at most M trials is $P = 1 - (1 - P')^M$, where P' is the probability of finding $\mathbf{f}(\mathbf{g})$ in one trial. Although P' is an unknown constant that depends on the size and shape of $\mathcal{S}_{\mathbf{o}'}$'s boundary, P tends to 1 exponentially in M for all $P' > 0$. So by making M sufficiently large, we can be confident that failure to find any $\mathbf{f}(\mathbf{g})$ after M trials means that there are very few similarity transformations \mathbf{f} such that $\mathcal{S}_{\mathbf{o}'}$ contains $\mathbf{f}(\mathbf{g})$. The most interesting cases are those where P' is small, which occur when the size of $\mathcal{S}_{\mathbf{o}'}$'s boundary is similar to the range of sizes of the boundaries of \mathbf{g} that we are sampling over. In our tests we used $M = 750$, which leads to a 99.95% probability of success within M trials when P' is 1%.

5.3.4 PCA-Based Deformation Mode Models

Once we have constructed a set of similarity-transformed canonical grids $\mathbf{f}(\mathbf{g})$ and warped them with the deformation fields as described in algorithm 5.1 to form sequences g'_{T_s} , the next steps are to calculate the affine registration functions h_t that remove the affine components from each g'_t and then to model the principal modes of residual deformation. The temporal ordering of the g'_t s is unimportant as far as modelling the deformation modes is concerned, so we will drop the time subscripts for most of the rest of this section.

Many approaches to deformation modelling have been proposed in the shape analysis literature, and the closest to our work is the Point Distribution Model (PDM) proposed by Cootes et al. in [22], which forms the basis of other important models of deformable objects, such as the

Active Appearance Model (AAM) [23]. The PDM in its original form is not quite applicable to our task however, as its training shapes must be defined in terms of landmarks whose correspondences across multiple shapes can be identified. In contrast, we need to be able to model the deformations of any visible part of the myocardium (as we explained earlier), and we do not have a reliable way of identifying corresponding 2D regions of the myocardium across patients (nor is it clear whether or not such correspondences can even be meaningfully defined for all parts of the myocardium). In addition, in the training phase of the PDM, only similarity transformations are removed from the training data, which means that the PDM learns skewing information. Again, as stated earlier, we do not want our deformation mode models to encode any information about the affine components of the deformations that the deformation fields induce on $\mathbf{f}(\mathbf{g})$, since we already have a parameterisation of affine transformations (eq. (3.1)).

We will begin by treating each $g = h(g')$ as an instantiation of a d -dimensional Euclidean random vector \mathbf{g} , with each component of \mathbf{g} corresponding to an ordinate of one of g 's vertices. For this representation of g , we use the unconstrained affine registration procedure described in section 5.3.2 to calculate h , and we assume that there is an orthogonal change of basis under which \mathbf{g} 's components undergo independent standard Gaussian variations, i.e. we assume that \mathbf{g} satisfies

$$\mathbf{g} = \boldsymbol{\mu}_g + \mathbf{V} \boldsymbol{\Lambda}^{\frac{1}{2}} \begin{pmatrix} \mathbf{x} \\ \boldsymbol{\epsilon} \end{pmatrix}, \quad (5.44)$$

where $\boldsymbol{\mu}_g$ is \mathbf{g} 's mean state, \mathbf{V} is a matrix of orthonormal basis vectors, $\boldsymbol{\Lambda}^{\frac{1}{2}}$ is a diagonal matrix denoting the standard deviations of $(\mathbf{g} - \boldsymbol{\mu}_g)$'s projections onto \mathbf{V} 's columns and \mathbf{x} and $\boldsymbol{\epsilon}$ are vectors of i.i.d. standard Gaussian random variables. We use the usual convention of the standard deviations decreasing monotonically from the top-left element of $\boldsymbol{\Lambda}^{\frac{1}{2}}$ to the bottom-right element, so that for any c , the first c columns of \mathbf{V} denote the c directions in which \mathbf{g} varies the most – the principal deformation modes. Examples of the principal modes for this Euclidean representation of g are given in figure 5.15 on page 168.

Our intention, in splitting the Gaussian variables between the n -dimensional subvector \mathbf{x} and the $(d-n)$ -dimensional subvector $\boldsymbol{\epsilon}$, is to capture pure noise in \mathbf{g} caused by errors in the deformation

field estimation process in the subspace $\boldsymbol{\mu}_g + \mathbf{V}_{:,n+1:d} \boldsymbol{\Lambda}_{n+1:d,n+1:d}^{\frac{1}{2}} \boldsymbol{\epsilon}$, and variations in \mathbf{g} mostly due to the true deformations in the orthogonal subspace $\boldsymbol{\mu}_g + \mathbf{V}_{:,1:n} \boldsymbol{\Lambda}_{1:n,1:n}^{\frac{1}{2}} \mathbf{x}$. This way, after estimating $\boldsymbol{\mu}_g$, \mathbf{V} and $\boldsymbol{\Lambda}^{\frac{1}{2}}$, we will be able to discard the noisy subspace, leaving a cleaner, lower-dimensional model of \mathbf{g} expressed in terms of \mathbf{x} . Note that due to the orthogonality of these subspaces, this equation is equivalent to the factor analysis model:

$$\mathbf{g} = \boldsymbol{\mu}_g + \mathbf{V}_{:,1:n} \boldsymbol{\Lambda}_{1:n,1:n}^{\frac{1}{2}} \mathbf{x} + \mathbf{V}_{:,n+1:d} \boldsymbol{\Lambda}_{n+1:d,n+1:d}^{\frac{1}{2}} \boldsymbol{\epsilon} , \quad (5.45)$$

in which \mathbf{x} is called the *common factor*, or *latent variable*. We will use the latter term. More general factor analysis problems with Gaussian latent variables are discussed in [62].

This model of \mathbf{g} 's distribution is closely related to the Gaussian factor analysis model with isotropic noise analysed by Tipping and Bishop in [111]. Their model has the following form:

$$\mathbf{g} = \boldsymbol{\mu}_g + \mathbf{W} \mathbf{y} + \boldsymbol{\epsilon}' , \quad (5.46)$$

where $\boldsymbol{\mu}_g + \mathbf{W} \mathbf{y}$ captures the noise-free variations in \mathbf{g} with a matrix \mathbf{W} , which is to be estimated from the data, and a k -dimensional standard Gaussian random vector \mathbf{y} . $\boldsymbol{\epsilon}'$ is a zero-mean isotropic Gaussian noise vector with variance σ^2 in each direction. In that paper, it was shown that the maximum-likelihood estimate of this model's parameters is given by a minor variant of the popular PCA (Principal Component Analysis) method (used for example by Coates et al. to define the PDM, and discussed in a model-free context in [54]). In particular, if we let \mathbf{G} be an M -column matrix, each column denoting a sample of \mathbf{g} , and we use $\bar{\mathbf{g}}$ and \mathbf{C} to denote the maximum-likelihood estimates of $\boldsymbol{\mu}_g$ and \mathbf{g} 's covariance matrix Σ , which are given by

$$\bar{\mathbf{g}} = \frac{1}{M} \sum_{j=1}^M \mathbf{G}_{:,j}, \quad \mathbf{C} = \frac{1}{M} \mathbf{G} \mathbf{G}^T - \bar{\mathbf{g}} \bar{\mathbf{g}}^T , \quad (5.47)$$

then the maximum-likelihood estimates σ'^2 and \mathbf{W}' of σ^2 and \mathbf{W} , respectively, are:

$$\sigma'^2 = \frac{1}{d-k} \sum_{i=k+1}^d \Lambda'_{i,i}, \quad \mathbf{W}' = \mathbf{V}'_{:,1:k} (\boldsymbol{\Lambda}'_{1:k,1:k} - \sigma'^2 \mathbf{I})^{\frac{1}{2}} , \quad (5.48)$$

where

$$\mathbf{C} = \mathbf{V}'\mathbf{\Lambda}'\mathbf{V}'^T \quad (5.49)$$

is the eigendecomposition of \mathbf{C} .

The main difference between this model and the one we gave in eq. (5.44) is that while our model does not account for errors in the subspace of the latent variables, it does account for anisotropic errors in the orthogonal subspace, whereas the Tipping and Bishop model accounts for errors in all directions, but assumes them to be isotropic. In practice, some noise will of course remain in the latent variable subspace with our model, and so in section 5.4 we will consider more effective ways of filtering out the remaining noise based on temporal smoothness assumptions. But for now, by setting $\sigma'^2 = \sigma^2 = 0$, $k = d$ and $\mathbf{y} = (\mathbf{x}^T, \boldsymbol{\epsilon}^T)^T$, it follows from Tipping and Bishop's work that the maximum-likelihood estimate of \mathbf{V} and $\mathbf{\Lambda}^{\frac{1}{2}}$ in our model are also \mathbf{V}' and $\mathbf{\Lambda}'$, which is the solution given by standard PCA.

It is worth noting that the standard maximum-likelihood estimators of $\boldsymbol{\mu}_g$ and $\boldsymbol{\Sigma}$ given above assume that the samples $\mathbf{G}_{:,j}$ are independent. Despite the \mathbf{g} s in each non-affine sequence g_{T_s} that we use to define \mathbf{G} not being independent of each other, the cross-validation tests we performed in section 5.5.1 suggest that it is acceptable to use all of the \mathbf{g} s in each sequence when estimating these parameters.

After calculating $\bar{\mathbf{g}}$, \mathbf{V}' and $\mathbf{\Lambda}'$ and selecting the number n of principal components to use, denoised estimates $\tilde{\mathbf{g}}$ of each instance \mathbf{g} of \mathbf{g} are easily constructed by projecting \mathbf{g} into \mathbf{x} 's subspace, and then mapping that projection back into the original space:

$$\tilde{\mathbf{g}} = \bar{\mathbf{g}} + \mathbf{V}'_{:,1:n}(\mathbf{V}'_{:,1:n})^T(\mathbf{g} - \bar{\mathbf{g}}) \quad (5.50)$$

We will discuss the selection of n in section 5.5.

5.3.5 Polar Coordinate PCA-Based Deformation Mode Models

The problem with applying PCA to this Euclidean representation of g is that it is difficult to justify the assumption that g 's ordinates exhibit linear dependencies on each other and Gaussian variations. Since the principal modes of the sample covariance matrix define a basis over the space of \mathbf{g} , we can always reconstruct each instance of \mathbf{g} as accurately as we like by using more modes. But it is possible that non-linear methods will lead to more compact models, that require less parameters to construct \mathbf{g} to any given degree of accuracy.

Numerous non-linear alternatives to PCA have been proposed in the literature, such as: kernel PCA [95], which exploits the “kernel trick” to map data samples to a higher-dimensional (potentially infinite-dimensional) space in which the dependencies are closer to linear, so that performing standard PCA in this space yields a more efficient representation; Locally Linear Embedding [88], which locally reconstructs data samples as linear combinations of their neighbours, and then uses eigenanalysis of a matrix defined in terms of the linear combination coefficients to find a lower-dimensional, neighbourhood-preserving parameterisation of the data; and Isomap [108], which uses the Euclidean distances between neighbouring data samples to determine the geodesic distances between all pairs of samples along the manifold that they are implicitly embedded in, and then uses eigenanalysis of a matrix defined in terms of the geodesic distances to calculate lower-dimensional representations.

The main problem with these and other related methods is their time and space complexity. Whereas standard PCA only requires two iterations over the M data samples, followed by the $O(d^3)^*$ eigendecomposition of a $d \times d$ covariance matrix, where d is the dimensionality of the sample space, all three of these methods involve computing eigendecompositions of $M \times M$ matrices. Our tests at the end of this chapter are based on two image sequences; one is 51 frames long, and the other is 59 frames long. For each test, we apply 1000 similarity transformations to the canonical grids, which means that we use $M = 51000$ and $M = 59000$. Storing $M \times M$ matrices for such large values of M would require 20–28GB of memory at double floating-point precision, which is far more memory than we have. And even if we could store such large

*When using the symmetric QR algorithm in [40].

matrices, the $O(M^3)$ eigendecompositions would be infeasible to carry out on current hardware.

Interestingly, [58] suggests a solution to the problem of computing the eigendecomposition for large M in the context of kernel PCA based on the Generalised Hebbian Algorithm, which was introduced through the work published in [75] and [92] as a way of training linear neural networks to find the eigenvectors of the autocorrelation matrix of their input samples. It may perhaps be possible to adapt this to other dimensionality reduction algorithms, and stream sections of the $M \times M$ matrix into memory as they are needed, but this would be a rather complex undertaking, and would probably still be very slow.

In light of these problems, we decided to test the simpler alternative of explicitly non-linearly mapping the vertices of the registered grids $g = h(g')$ to another space, applying linear PCA in this space, and then comparing PCA's performance in this space to its performance in Euclidean space. One of the simplest non-linear mappings to try is one in which we map the vertices of the gs defined by deformations of the quasi-circular canonical grid \mathbf{g}_c to a polar coordinate space. We use this mapping under the assumption that the myocardial surface can be modelled as a surface composed of small regions that undergo local rotations and stretch radially away from or towards their rotational centres. So for this space, we use the constrained affine mapping defined in section 5.3.2 as the affine registration function h , enforcing the constraint that h should map the centres of the deformed grids g' to the origin, so as to yield a consistent mapping.

After computing the hs , we represent each registered g with a $2sr$ -dimensional vector \mathbf{g} (s and r are the number of sectors and rings, as before). Given some ordering (r_i, s_i) ($i \in 1, \dots, sr$) over the (ring,sector) indices of g 's vertices, we set \mathbf{g} 's even ordinates to the radial distances of g 's vertices from the origin:

$$\mathbf{g}_{2i} = \|g(r_i, s_i)\| \quad , \quad (5.51)$$

and each odd ordinate \mathbf{g}_{2i-1} to a scaling factor k_i multiplied by $g(r_i, s_i)$'s angular deviation

from the mean direction of the i^{th} vertex of all gs in the data set:

$$\mathbf{g}_{2i-1} = \begin{cases} 0 & , r_i = 0 \\ k_i \arcsin \left(g^*(r_i, s_i) \times \frac{g(r_i, s_i)}{\|g(r_i, s_i)\|} \right) & , \text{otherwise} \end{cases} \quad (5.52)$$

where

$$\|g^*(r_i, s_i)\| = 1 \quad , \quad g^*(r_i, s_i) \propto \frac{1}{T|G_S|} \sum_{(\cdot, \cdot, \cdot, g) \in G_S} \sum_{t=1}^T \frac{g_t(r_i, s_i)}{\|g_t(r_i, s_i)\|} \quad . \quad (5.53)$$

The space that these \mathbf{g} s reside in can be considered to be locally Euclidean, provided that the angular deviations lie within the range $(-\frac{\pi}{2}, \frac{\pi}{2})$ radians (in our tests they were well within these bounds). So, taking each \mathbf{g} to be an instantiation of a Gaussian random vector, we use linear PCA to construct deformation mode models in this space, exactly as described in section 5.3.4. Since PCA identifies directions of maximum variance, it may not give meaningful results when using data sets that consist of different types of measurements (such as lengths and angles). So we use the factor k_i to scale the angular deviations to a magnitude comparable to that of the radial lengths. We tried two different scaling methods, both based on the fact that the length of a circular arc that subtends an angle ϕ radians is given by $l\phi$, where l is the arc's radius. For the first method, we scaled all angles uniformly, setting k_i to the mean radial distance of the vertices from the origin, calculated over all vertices of all gs :

$$k_i = \frac{1}{srT|G_S|} \sum_{(\cdot, \cdot, \cdot, g) \in G_S} \sum_{t=1}^T \sum_{i=1}^{sr} \|g(r_i, s_i)\| \quad \forall i \quad . \quad (5.54)$$

This preserves the fact that under the polar coordinate system, angular deviations of δ_ϕ in two vertices at different distances from the origin are considered to be equally significant. The second method comes from a suggestion in the literature, which we will mention at the end of this section.

Denoised estimates $\tilde{\mathbf{g}}$ of \mathbf{g} calculated as in eq. (5.50) will of course be in polar coordinates, but for analysis we prefer to convert each (radial length, scaled angular deviation) pair $(l_i, k_i \delta_{\phi, i})$ back to Cartesian coordinates, so that we can compare the results with those of the previous

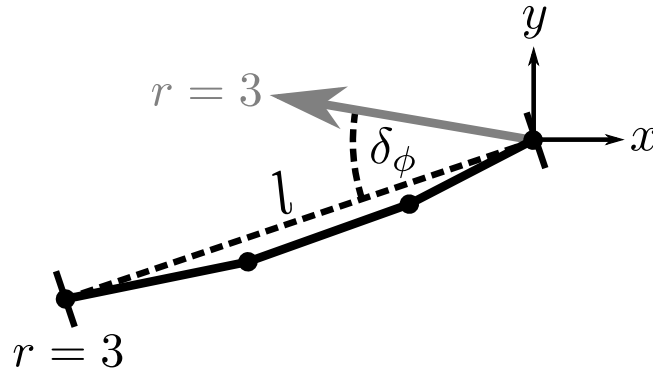


Figure 5.7: The ring 3 vertex of the spoke of g denoted by the solid black line is at a distance l from the origin, and it rotates away from the corresponding mean directional vector (shown in grey) by δ_ϕ radians.

standard PCA approach. We perform the conversion as follows:

$$\begin{pmatrix} x \\ y \end{pmatrix} = \begin{cases} \mathbf{0} & , r_i = 0 \\ l_i \begin{pmatrix} \cos(\delta_{\phi,i}) & -\sin(\delta_{\phi,i}) \\ \sin(\delta_{\phi,i}) & \cos(\delta_{\phi,i}) \end{pmatrix} g^*(r_i, s_i) & , \text{otherwise} . \end{cases} \quad (5.55)$$

We are not by any means the first to use polar coordinates in PCA. It was used, for example, by Heap and Hogg in [48] in a hybrid shape model for articulated objects like human bodies or anglepoise lamps. In their model, collections of points with strong rotational interdependencies were represented in polar coordinates, and the other points were represented in Cartesian coordinates. Besides the fact that we use constrained affine transformations to register our shapes rather than the more common Procrustes method that they use, the main difference between our use of polar coordinates and theirs is to do with the kind of angles they measure.

As their model was designed for articulated objects, they needed to identify pivots \mathbf{c} within each object. For each object point \mathbf{p} that rotates about \mathbf{c} , rather than measuring \mathbf{p} 's angular deviation from a fixed reference direction as we do with g^* , they measured the angle between $\overrightarrow{\mathbf{c}\mathbf{p}}$ and $\overrightarrow{\mathbf{c}\mathbf{x}}$ for some reference point \mathbf{x} that is also on the object. E.g. if \mathbf{p} is a point on a human wrist and \mathbf{c} is on the elbow, then the reference point might be chosen to be somewhere on the shoulder, and the measured angle would be roughly equal to the angle between the upper arm and the forearm. Measuring angles like this means that the order in which points are

reconstructed from linear combinations of principal components is important; before \mathbf{p} can be reconstructed, its reference point and \mathbf{c} must be reconstructed. In our case however, such an angle measurement scheme would have created a cycle of dependencies around the vertices of each ring, making reconstruction an underdetermined task.

The second method for calculating k_i that we tried was the method that they suggested, of multiplying the angle measured for each (\mathbf{p}, \mathbf{c}) pair by $\|\mathbf{p} - \mathbf{c}\|$, which in our case gives

$$k_i = \|\mathbf{g}(r_i, s_i)\| . \quad (5.56)$$

They chose this scaling method because they considered it to be important to preserve the fact that if \mathbf{p}' and \mathbf{p} both rotate about \mathbf{c} by the same angle, then the point which is furthest from \mathbf{c} moves the furthest. We will present results using both scaling methods at the end of this chapter.

5.4 Deformation Trajectory Mode Modelling

Now that we have described our models of the static poses that we can expect the canonical grids to be in as a result of the non-affine components of myocardial deformation, the final stage is to incorporate these models into a model that takes into account not only the temporal dependencies between grids that define a sequence of deformations, but also the dependencies that exist between the affine components of the deformations and the non-affine components. The fact that there are strong dependencies between the affine and non-affine components should be clear from considering, for example, how elastic surfaces undergoing local twisting often also undergo some rotation, or how elastic surfaces that are stretched tend to thin in the middle. So we will begin this section with a formal definition of what we mean by a “deformation trajectory model”. We will then make their link to the deformation mode models of the previous section clear, and finally we will describe the specific types of deformation trajectory model that we have constructed.

In the previous section, we considered the affine registration functions h_{T_s} and the non-affine canonical grid deformations g_{T_s} to be deterministic, since they were fixed given: a deformation field sequence ν_{T_s} , a random similarity transformation \mathbf{f} and a random phase offset \mathbf{o} . We will now change our perspective by discarding these dependencies, so that the deterministic tuple $(h_{T_s}, g_{T_s}) \mid (\nu_{T_s}, \mathbf{f}, \mathbf{o})$ becomes the marginalised random tuple $(\mathbf{h}_{T_s}, \mathbf{g}_{T_s})$ that is the result of applying the sampling procedure described in section 5.3.3 to a deformation field sequence uniformly sampled from a training set V . Extending the previous section's notation, we will use the random reference frame \mathbf{t}_r , random cardiac cycle duration \mathbf{T} and random frame list $\mathbf{T}_s = [\mathbf{t}_r, \dots, \mathbf{t}_r + \mathbf{T} - 1]$ to denote the frames over which $(\mathbf{h}_{T_s}, \mathbf{g}_{T_s})$ is defined.

By the laws of large numbers, as we increase the size of V by defining deformation field sequences over more patients, the empirical distribution of the samples we draw of $(\mathbf{h}_{T_s}, \mathbf{g}_{T_s})$ should approach the true distribution over all patients of the sequence of non-affine and affine deformations that may occur in a region of the myocardium covered by a randomly similarity-transformed canonical grid (provided, of course, that the errors in the deformation field estimation procedure are not too large).

We will use $(\mathcal{H}\mathcal{G})$ to denote the space over which the physically plausible instances of $(\mathbf{h}_t, \mathbf{g}_t)$ vary (over all patients), for any frame \mathbf{t} , and we will refer to a particular tuple in this space as (h, g) . In addition, we will also use $(\mathcal{H}\mathcal{G})^{\mathcal{T}}$ to denote the space of all physically plausible random sequence tuples $(\mathbf{h}_{T_s}, \mathbf{g}_{T_s})$.

5.4.1 Deformation Trajectory Models

Let $(\mathcal{Q}_n, \langle \cdot, \cdot \rangle_{\mathcal{Q}})$ be an inner product space, where \mathcal{Q}_n is a vector space of continuous, periodic paths $\mathbf{q} : \mathbb{R} \rightarrow \mathbb{R}^n$ through Euclidean n -dimensional space, each with unit period, and $\langle \cdot, \cdot \rangle_{\mathcal{Q}}$ is an inner product on that space. We require the degree of parametric continuity of each path at integer parameters to be no less than the minimum degree of parametric continuity at all other parameter values, so that the join between consecutive cycles will be as smooth as the

rest of the path. So each $\mathbf{q} \in \mathcal{Q}_n$ satisfies

$$\left. \frac{d^k \mathbf{q}(t)}{dt^k} \right|_{t=s} = \left. \frac{d^k \mathbf{q}(t)}{dt^k} \right|_{t=s \pmod{1}} \quad (5.57)$$

for all $s \in \mathbb{R}$ and integers $k \in \{0, \dots, k'\}$, where k' is the smallest integer such that for some $s \in (0, 1)$,

$$\left. \frac{d^{k'+1} \mathbf{q}(t)}{dt^{k'+1}} \right|_{t=s} \quad (5.58)$$

is undefined. Note that k' may be infinite if the paths are infinitely differentiable everywhere.

We will use the paths of \mathcal{Q}_n to model the sequences of affine and non-affine deformations that the similarity-transformed canonical grids undergo. The periodicity and differentiability requirements ensure that we can begin each sequence at any phase without encountering sudden “jumps” in the middle.

Let Υ^+ be a continuous mapping from \mathbb{R}^n to some superset of $(\mathcal{H}\mathcal{G})$. We require Υ^+ to be one-to-one, and hence to have an inverse, which we will call Υ . Thus, $\Upsilon(\Upsilon^+(\mathbf{x})) = \mathbf{x}$ for all $\mathbf{x} \in \mathbb{R}^n$. Υ need not be one-to-one, but we require Υ^+ to be its pseudoinverse (in the sense that for some appropriate measure of distance, $\Upsilon^+(\Upsilon(h, g)) = (h', g')$ is the closest tuple to (h, g) in Υ^+ 's image). Υ is the function we will use to perform an initial noise-filtering dimensionality reduction on (h, g) , and so it may discard information about its argument, but the reconstruction function Υ^+ must, of course, preserve all information about its argument.

We model the sequence $(\mathbf{h}_{\mathbf{T}_s}, \mathbf{g}_{\mathbf{T}_s})$ as a noisy sample of Υ^+ 's image of some random path $\mathbf{q} \in \mathcal{Q}_n$, in the sense that

$$(\mathbf{h}_t, \mathbf{g}_t) = \epsilon_t \left(\Upsilon^+ \left(\mathbf{q} \left(\frac{t - \mathbf{t}_r}{\mathbf{T}} \right) \right) \right) , \quad (5.59)$$

where the $\epsilon_{\mathbf{t}_r}, \dots, \epsilon_{\mathbf{t}_r + \mathbf{T} - 1}$ are random perturbation functions that result from errors in the estimation of the deformation fields and deviations from periodicity due to unmodelled factors such as the influence of respiratory motion. Whatever model is chosen for the distribution of the ϵ_t , we require them to satisfy

$$E[(\mathbf{h}_t, \mathbf{g}_t) \mid \mathbf{q} = \mathbf{q}] = \Upsilon^+ \left(\mathbf{q} \left(\frac{t - \mathbf{t}_r}{\mathbf{T}} \right) \right) , \quad (5.60)$$

so that they have no effect on average.

Definition 5.1. *Given \mathcal{Q}_n and Υ^+ for which there are random paths $\mathbf{q} \in \mathcal{Q}_n$ that induce the distribution of the $(\mathbf{h}_{\mathbf{T}_s}, \mathbf{g}_{\mathbf{T}_s})$ up to observational noise in the sense of eq. (5.59), we call each particular instance \mathbf{q} of \mathbf{q} a **deformation trajectory**.*

Definition 5.2. *A **deformation trajectory model** (DTM) is a tuple*

$$(\mathcal{Q}_n, \langle \cdot, \cdot \rangle_{\mathcal{Q}}, \Upsilon^+, \Xi_{\mathcal{Q}}, \Xi_{\epsilon}) , \quad (5.61)$$

where:

- $\mathcal{Q}_n, \langle \cdot, \cdot \rangle_{\mathcal{Q}}$ and Υ^+ are defined as above;
- $\Xi_{\mathcal{Q}}$ is a probability space over \mathcal{Q}_n defining the distribution of the deformation trajectories;
- Ξ_{ϵ} is a probability space defining the distribution of the observational noise functions ϵ_t .

Once a DTM has been chosen, the main tasks are to infer the parameters of $\Xi_{\mathcal{Q}}$'s probability measure from our noisy observations of $(\mathbf{h}_{\mathbf{T}_s}, \mathbf{g}_{\mathbf{T}_s})$, and then to analyse the principal deformation trajectory modes of the parameterised distribution (i.e. the principal ways in which the observed $(\mathbf{h}_{\mathbf{T}_s}, \mathbf{g}_{\mathbf{T}_s})$ sequences vary). We shall make all of this concrete in the following sections, in which we will consider a simple, but effective, type of DTM.

5.4.2 B-Spline Inner Product Spaces

B-splines curves provide a particularly convenient model for the multidimensional paths of a DTM. We shall describe their properties in sufficient detail to make our use of them clear, but for more general information about them, we refer the reader to one of the many published references, such as [26] and [81].

Let $S_{n,c,d}$ be the vector space of periodic B-splines in \mathbb{R}^n for which every B-spline: has the same degree $d \geq 1$; has the same number of control points $c \geq 2d$; is defined on the uniform sequence

of $c + d + 1$ knots:

$$t_i = \frac{i - d - 1}{c - d}, \quad 1 \leq i \leq c + d + 1, \quad (5.62)$$

so that $t_{d+1} = 0$ and $t_{c+1} = 1$. We enforce periodicity by requiring the first d control points to be equal to the last d . These requirements mean that all B-splines in the space are defined on the interval $[0, 1]$, have order $d - 1$ parametric continuity everywhere in it, and, by virtue of their periodicity, can be evaluated at all real parameters by taking the parameters modulo 1, as in eq. (5.57).

$S_{n,c,d}$'s B-splines are uniquely defined by nc -vectors, which we shall refer to as the *trajectory vectors*. Each trajectory vector \mathbf{p} is the vector form of an $n \times c$ matrix $[\mathbf{p}]$, the columns of which define the B-spline control points:

$$[\mathbf{p}] \triangleq \begin{pmatrix} \mathbf{p}_1 & \cdots & \mathbf{p}_c \\ \mathbf{p}_{c+1} & \cdots & \mathbf{p}_{2c} \\ \vdots & \vdots & \vdots \\ \mathbf{p}_{(n-1)c+1} & \cdots & \mathbf{p}_{nc} \end{pmatrix}. \quad (5.63)$$

For any given \mathbf{p} , the B-spline $\mathbf{q}(t \in [0, 1]; \mathbf{p})$ is defined as

$$\mathbf{q}(t; \mathbf{p}) = [\mathbf{p}] \mathbf{b}(t), \quad (5.64)$$

where $\mathbf{b}(t) = (b_{d,1}(t), \dots, b_{d,c}(t))^T$ is a vector of degree d B-spline basis functions, which can be defined by the recursive Cox-de Boor equations:

$$b_{0,i}(t) = \begin{cases} 1 & , t \in [t_i, t_{i+1}) \\ 0 & , \text{otherwise,} \end{cases}$$

$$b_{n,i}(t) = \frac{t - t_i}{t_{i+n} - t_i} b_{n-1,i}(t) + \frac{t_{i+n+1} - t}{t_{i+n+1} - t_{i+1}} b_{n-1,i+1}(t). \quad (5.65)$$

It is trivial to show that $S_{n,c,d}$ is a vector space. It suffices to consider the fact that linear

combinations of any two B-splines $\mathbf{q}(\cdot; \mathbf{p})$ and $\mathbf{q}(\cdot; \mathbf{p}')$ in $S_{n,c,d}$ satisfy

$$\begin{aligned} \alpha \mathbf{q}(\cdot; \mathbf{p}) + \beta \mathbf{q}(\cdot; \mathbf{p}') &= \alpha [\mathbf{p}] \mathbf{b}(\cdot) + \beta [\mathbf{p}'] \mathbf{b}(\cdot) \\ &= (\alpha [\mathbf{p}] + \beta [\mathbf{p}']) \mathbf{b}(\cdot) , \end{aligned} \quad (5.66)$$

which is the B-spline defined by the trajectory vector $\alpha \mathbf{p} + \beta \mathbf{p}'$. Hence, all of the vector space axioms can be expressed in terms of the trajectory vectors (or equivalently, the control points), which themselves obviously define a vector space (more specifically, they define a subspace of \mathbb{R}^{nc} , since the first d control points are repeated). So for some choice of c and d , we can set $\mathcal{Q}_n = S_{n,c,d}$.

B-Spline PCA

We will now momentarily side-step the question of what probability distribution to impose upon $S_{n,c,d}$ and consider how PCA may be extended to the analysis of a set of B-splines in a distribution-free setting.

Whatever inner product $\langle \cdot, \cdot \rangle$ is chosen, the problem of identifying linear principal components can be expressed in much the same way as the usual distribution-free finite-dimensional Euclidean formulation. In particular, given a set of m sample B-splines $\mathbf{q}(\cdot; \mathbf{p}^{[i]}) \in S_{n,c,d}$ with sample mean $\bar{\mathbf{q}}(\cdot) = \mathbf{q}(\cdot; \bar{\mathbf{p}})$, where

$$\bar{\mathbf{p}} = \frac{1}{m} \sum_{i=1}^m \mathbf{p}^{[i]} , \quad (5.67)$$

the first k principal components are functions $\boldsymbol{\vartheta}^{[j]} : [0, 1] \rightarrow \mathbb{R}^n$ that each maximise the variance of

$$\left\langle \boldsymbol{\vartheta}^{[j]}, \mathbf{q}(\cdot; \mathbf{p}^{[i]}) - \bar{\mathbf{q}} \right\rangle \quad (5.68)$$

over all i , subject to the orthonormality condition

$$\left\langle \boldsymbol{\vartheta}^{[i]}, \boldsymbol{\vartheta}^{[j]} \right\rangle = \delta_{i,j} , \quad \forall (i, j) \in \{1, \dots, k\}^2 . \quad (5.69)$$

The B-splines can then be reconstructed (to a degree of accuracy that increases with k) by

calculating linear combinations of the $\boldsymbol{\vartheta}^{[j]}$:

$$\mathbf{q}(t; \mathbf{p}^{[i]}) \approx \bar{\mathbf{q}}(t) + \sum_{j=1}^k \left\langle \boldsymbol{\vartheta}^{[j]}, \mathbf{q}(\cdot; \mathbf{p}^{[i]}) - \bar{\mathbf{q}} \right\rangle \boldsymbol{\vartheta}^{[j]}(t) . \quad (5.70)$$

Defining \mathbf{W}^- as an $n(c-d) \times nc$ block diagonal matrix that discards the duplicate trajectory vector dimensions, i.e.

$$\begin{aligned} \mathbf{W}^- &\triangleq \text{diag}(\overbrace{\widehat{\mathbf{W}}^- \quad \dots \quad \widehat{\mathbf{W}}^-}^{\times n}) \\ \widehat{\mathbf{W}}^- &\triangleq \begin{pmatrix} \mathbf{I}_{(c-d)} & \mathbf{0}_{(c-d) \times d} \end{pmatrix} , \end{aligned} \quad (5.71)$$

a simple choice of inner product is the Euclidean inner product $\mathbf{p}'^T \mathbf{W}^{-T} \mathbf{W}^- \mathbf{p}$, for arbitrary trajectory vectors \mathbf{p} and \mathbf{p}' . For this choice, the above method reduces to mapping each sample B-spline $\mathbf{q}(\cdot; \mathbf{p}^{[i]})$ to the point $\mathbf{W}^- \mathbf{p}^{[i]}$, and performing PCA on these points as described earlier. The j^{th} principal component will be a B-spline in $S_{n,c,d}$ with a trajectory vector $\mathbf{v}^{[j]}$ defined such that $\mathbf{W}^- \mathbf{v}^{[j]}$ is the j^{th} largest eigenvector of the sample trajectory vector covariance submatrix \mathbf{C}^* , defined in terms of the full covariance matrix \mathbf{C} by removing duplicate rows and columns, i.e.:

$$\begin{aligned} \mathbf{C}^* &= \mathbf{W}^- \mathbf{C} \mathbf{W}^{-T} , \\ \mathbf{C} &= \frac{1}{m} \sum_{i=1}^m \mathbf{p}^{[i]} \mathbf{p}^{[i]T} - \bar{\mathbf{p}} \bar{\mathbf{p}}^T . \end{aligned} \quad (5.72)$$

The remaining dimensions of $\mathbf{v}^{[j]}$ are defined by duplication, to make the corresponding B-spline periodic.

A more common choice of inner product for vector-valued functions over $[0, 1]$ is the sum of L^2 inner products over the dimensions of the range, which for B-splines in $S_{n,c,d}$ is

$$\langle \mathbf{q}(\cdot; \mathbf{p}), \mathbf{q}(\cdot; \mathbf{p}') \rangle_{L^2}^+ \triangleq \sum_{i=1}^n \int_0^1 \mathbf{q}_i(t; \mathbf{p}) \mathbf{q}_i(t; \mathbf{p}') dt = \int_0^1 \mathbf{q}(t; \mathbf{p})^T \mathbf{q}(t; \mathbf{p}') dt . \quad (5.73)$$

This inner product was used with more general functions in [18, 85], and the latter of these

publications gave a closed-form method for computing the eigenfunctions $\boldsymbol{\vartheta}^{[j]}$ when the sample functions have a finite basis expansion. For periodic B-splines, the method involves convolving the columns of \mathbf{C}^* (with wraparound) with a kernel defined in terms of the basis functions \mathbf{b} , and then performing PCA as normal on the result to determine the eigenvalues and the trajectory vectors of the eigenfunctions (which are B-splines). The resulting matrix is asymmetric, but the problem can be transformed into a more computationally stable symmetric eigenanalysis problem. The kernel is symmetric and decreases away from its centre. For the degree 3 B-splines that we use, it is also very narrow and about half of its total weight is concentrated at its centre, so the resulting eigenfunctions should not be very different to the result given by the Euclidean inner product above. So we use the simpler Euclidean inner product instead.

5.4.3 A B-Spline-Based Gaussian DTM

Taking $S_{n,c,d}$ as \mathcal{Q}_n and the Euclidean inner product of the control points as $\langle \cdot, \cdot \rangle_{\mathcal{Q}}$, we must now consider the construction of the space \mathbb{R}^n that the deformation trajectories reside in. Under our model, the path that each deformation trajectory follows through this space is a representation of some continuous sequence of affine and non-affine transformations that we can only observe indirectly at discrete time points in the form of the noisy $(\mathbf{h}_{T_s}, \mathbf{g}_{T_s})$ tuples. So for the Euclidean inner product over the trajectory vectors to be a meaningful choice of $\langle \cdot, \cdot \rangle_{\mathcal{Q}}$, the space of the representations of the transformations that we choose must be as close to Euclidean as possible.

Our assumption in defining the different types of deformation mode model that we considered earlier was that at least one of the types would be a reasonable approximation of a Euclidean space of non-affine deformations. So given a deformation mode model that reduces the number of dimensions of each non-affine grid \mathbf{g}_t to m , we define the first m dimensions of \mathbb{R}^n as the subspace of projections of the \mathbf{g}_t onto the first m principal components of the deformation mode model, turning the DTMs we create into nested principal component models. We will use $\tilde{\mathbf{g}}_{T_s}$ to denote these random m -dimensional projections, and $\tilde{\mathbf{g}}_{T_s}$ for specific instances of them.

As we mentioned in chapter 3, matrix logarithms map matrices to locally Euclidean spaces, and so they are good candidates for approximately Euclidean representations of the affine

deformation components. For the application of the DTMs to deformation simulation that we will discuss in the next chapter, it will be more useful for the DTMs to give a representation of the inverted affine components $\mathbf{h}_{T_s}^{-1}$, since these map from the coordinate system of the canonical grids to that of the deformation fields. In addition, we will not need to model the translational components of the $\mathbf{h}_{T_s}^{-1}$, and we will only need to know the relative states of each pair of transformations in $\mathbf{h}_{T_s}^{-1}$. In other words, for each instance h_{T_s} of \mathbf{h}_{T_s} , it is acceptable for the DTM to give an approximation of $(h_{t_r}^{*-1} \circ h', \dots, h_{t_r+T-1}^{*-1} \circ h')$ as a reconstruction of $h_{T_s}^{-1}$, where h_t^{*-1} represents the inverse of the non-translational component of h_t , and h' is an arbitrary non-singular affine transformation that may be different for each instance of \mathbf{h}_{T_s} . The non-translational component of each 2D affine transformation h_t is a linear transformation with 4 degrees of freedom (c.f. the \mathbf{A} matrix in eq. (3.1)) that we will represent as the 2×2 matrix $[\mathbf{h}_t^*]$. So we set $n = m + 4$ and define the last 4 dimensions of \mathbb{R}^n to be the space in which the elements of the logarithms of these matrices reside.

We can now define the mapping Υ and its pseudoinverse Υ^+ . $\Upsilon(h, g)$ returns an $(m + 4)$ -vector \mathbf{x} such that

$$\mathbf{x}_{1:m} = \tilde{\mathbf{g}} \ , \quad (5.74)$$

and

$$\begin{pmatrix} \mathbf{x}_{m+1} & \mathbf{x}_{m+2} \\ \mathbf{x}_{m+3} & \mathbf{x}_{m+4} \end{pmatrix} = \log([\mathbf{h}^*]) \ . \quad (5.75)$$

We refer to \mathbf{x} as a **deformation vector**. Υ^+ then maps \mathbf{x} to the pair (h', g') , where the non-translational component of h' is given by the matrix exponential of the 2×2 matrix that $\mathbf{x}_{m+1:m+4}$ represents, the translational component is zero, and g' is the result of reconstructing $\mathbf{x}_{1:m}$ in the original space, as described in earlier sections.

For each $(\mathbf{h}_{T_s}, \mathbf{g}_{T_s})$, the fact that we only need to know the relative states of the inverted affine components allows us to transform them into a coordinate system in which the magnitudes of their rotational components are no larger than the range of rotations that occurs in $\mathbf{h}_{T_s}^{*-1}$, reducing the likelihood of the matrix logarithms being complex (see appendix B.2). We can accomplish this by simply composing each \mathbf{h}_t^{*-1} with $\mathbf{h}_{\check{t}}^*$, for some $\check{t} \in T_s$. The choice of \check{t} is

somewhat arbitrary, so we choose one that endows each sequence

$$(\mathbf{x}^{[t_r]}, \dots, \mathbf{x}^{[t_r+T-1]}) = (\Upsilon(\mathbf{h}_{t_r}^{*-1} \circ \mathbf{h}_{t_r}^*, \mathbf{g}_{t_r}), \dots, \Upsilon(\mathbf{h}_{t_r+T-1}^{*-1} \circ \mathbf{h}_{t_r}^*, \mathbf{g}_{t_r+T-1})) \quad (5.76)$$

of deformation vectors with a “watermark” that allows us to estimate the phase offset \mathbf{o} that was chosen when $(\mathbf{h}_{T_s}, \mathbf{g}_{T_s})$ was sampled. As we shall soon see, this watermark carries over to the deformation trajectories that model the deformation vector sequences, and hence to the deformation trajectories that we will sample from \mathcal{Q}_n when carrying out simulations in the next chapter, allowing the simulations to identify useful starting points in the deformation trajectories. This choice of $\check{\mathbf{t}}$ also makes the deformation vector sequences mutually consistent in the sense that it ensures that they lie in the same coordinate system, and hence have the same geometric significance.

The watermark relies on the following facts. Firstly, it follows from eq. (5.17) that $\mathbf{g}_{t_r+\mathbf{o}}$ is just the result of registering a similarity-transformed canonical grid $\mathbf{f}(\mathbf{g})$ to \mathbf{g} , and hence $\mathbf{g}_{t_r+\mathbf{o}} = \mathbf{g}$ for both of the registration methods that we have considered. Secondly, due to the quasi-uniform nature of the similarity transformation sampling process, the mean state of the non-affine grids \mathbf{g} should always tend to \mathbf{g} as the number of samples increases (see the top row of figure 5.15 for evidence of this). Hence given \mathbf{g}_{T_s} alone, we can identify \mathbf{o} by searching for the \mathbf{o}' such that $\mathbf{g}_{t_r+\mathbf{o}'} = \mathbf{g}$, and given $\mathbf{x}^{[T_s]}$ alone, we can estimate \mathbf{o} as the \mathbf{o}'' that minimises $\|\mathbf{x}_{1:m}^{[t_r+\mathbf{o}'']}\|^2$, since $\mathbf{x}_{1:m} = \tilde{\mathbf{g}} = \mathbf{0}$ corresponds to the deformation mode model’s mean. This estimate also minimises $\|\mathbf{x}^{[t_r+\mathbf{o}'']}\|^2$, since $\mathbf{x}_{m+1:m+4}^{[t_r+\mathbf{o}'']}$ is a vectorisation of

$$\log \left([\mathbf{h}_{t_r+\mathbf{o}''}^{*-1} \circ \mathbf{h}_{t_r+\mathbf{o}''}^*] \right) = \log(\mathbf{I}) = \mathbf{0} . \quad (5.77)$$

We extend this notion to the deformation trajectories with the following definition:

Definition 5.3. A value $\rho \in [0, 1)$ is a **candidate phase offset** for a deformation trajectory $\mathbf{q} \in \mathcal{Q}_n$ if and only if

$$\forall \rho' \in [0, 1) \left(\|\mathbf{q}(\rho)\|^2 \leq \|\mathbf{q}(\rho')\|^2 \right) . \quad (5.78)$$

The final parts of the DTM that we have to define are the distributions $\Xi_{\mathcal{Q}}$ and Ξ_{ϵ} over

the deformation trajectories and the observational noise, respectively. We assumed earlier, when defining our deformation mode models, that the \mathbf{g}_{T_s} could be represented in a (locally) Euclidean space in which they were corrupted by Gaussian noise, and that mapping them to the $\tilde{\mathbf{g}}_{T_s}$ vectors had the effect of removing some of this noise. Extending these assumptions to include the affine components, we assume that each $(\mathbf{h}_{T_s}, \mathbf{g}_{T_s})$ has a quasi-periodic subtrajectory $(\mathbf{h}_{T'_s}, \mathbf{g}_{T'_s})$, where

$$T'_s = [t_r, \dots, t_r + T' - 1], \quad 0 < T' \leq T, \quad (5.79)$$

such that each time point t of the corresponding deformation vector sequence $\mathbf{x}^{[T'_s]}$ (defined as in eq. (5.76)) satisfies

$$\mathbf{x}^{[t]} = \mathbf{q} \left(\frac{t - t_r}{T'}; \mathbf{p} \right) + \boldsymbol{\epsilon}^{[t]}, \quad (5.80)$$

for some random trajectory vector \mathbf{p} and zero mean Gaussian random vector $\boldsymbol{\epsilon}^{[t]}$. We allow the last $T - T'$ time points of $(\mathbf{h}_{T_s}, \mathbf{g}_{T_s})$ to be discarded because: a) we only approximately estimate the cardiac period when choosing the number of frames to define the deformation fields over; and b) due to the effects of respiratory motion and the non-uniform contraction of the heart, the times at which different regions of the myocardium return to states that are closest to their time t_r states vary. So we calculate T' by searching for the time step near the end of T_s at which the deformation vector sequence comes closest to its initial state $\mathbf{x}^{[t_r]}$:

$$T' = \arg \min_{T'' \in \{[0.75T], \dots, T-1, T\}} \|\mathbf{x}^{[t_r + T'' - 1]} - \mathbf{x}^{[t_r]}\|^2. \quad (5.81)$$

If we make the simplifying assumption that the $\boldsymbol{\epsilon}_t$ are i.i.d. vectors with diagonal covariance matrix $\boldsymbol{\Sigma} = \text{diag}(\sigma_1^2, \dots, \sigma_{m+4}^2)$, so that

$$\boldsymbol{\epsilon}_t \sim N(\boldsymbol{\epsilon}; \mathbf{0}, \boldsymbol{\Sigma}) \propto \exp \left\{ -\frac{1}{2} \boldsymbol{\epsilon}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\epsilon} \right\}, \quad (5.82)$$

then the maximum log-likelihood estimate of the trajectory vector \mathbf{p} coincides with the least

squares estimate, since

$$\begin{aligned}
\ln P(\mathbf{x}^{[T_s']} | \mathbf{p} = \mathbf{p}) &= \sum_{t=\mathbf{t}_r}^{\mathbf{t}_r+T'-1} \ln P\left(\boldsymbol{\epsilon}^{[t]} = \mathbf{x}^{[t]} - \mathbf{q}\left(\frac{t-\mathbf{t}_r}{T'}; \mathbf{p}\right)\right) \\
&= \sum_{t=\mathbf{t}_r}^{\mathbf{t}_r+T'-1} \ln N\left(\mathbf{x}^{[t]} - \mathbf{q}\left(\frac{t-\mathbf{t}_r}{T'}; \mathbf{p}\right); \mathbf{0}, \boldsymbol{\Sigma}\right) \\
&= - \sum_{t=\mathbf{t}_r}^{\mathbf{t}_r+T'-1} \sum_{i=1}^{m+4} \frac{1}{2} \left(\frac{\mathbf{x}_i^{[t]} - \mathbf{q}_i\left(\frac{t-\mathbf{t}_r}{T'}; \mathbf{p}\right)}{\sigma_i} \right)^2 + \text{constant} , \quad (5.83)
\end{aligned}$$

which can be maximised over each dimension of the B-spline independently, irrespective of the values of the standard deviations σ_i^2 . Reusing the control point duplicate removal matrix $\widehat{\mathbf{W}}^{-T}$, and defining $\widehat{\mathbf{W}}^+$ as a $c \times (c-d)$ matrix that duplicates the first d control points of a $(c-d)$ -dimensional control point vector:

$$\widehat{\mathbf{W}}^+ \triangleq \begin{pmatrix} \mathbf{I}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{I}_{(c-2d)} \\ \mathbf{I}_d & \mathbf{0} \end{pmatrix} , \quad (5.84)$$

the maximum log-likelihood solution is given by the control point matrix $[\mathbf{p}'] \widehat{\mathbf{W}}^{-T} \widehat{\mathbf{W}}^{+T}$ that minimises the squared error

$$\begin{aligned}
&\|\mathbf{X} - [\mathbf{p}'] \widehat{\mathbf{W}}^{-T} \widehat{\mathbf{W}}^{+T} \mathbf{B}\|_F^2 = \\
&\left\| \begin{pmatrix} \mathbf{x}^{[\mathbf{t}_r]} & \dots & \mathbf{x}^{[\mathbf{t}_r+T'-1]} \end{pmatrix} - [\mathbf{p}'] \widehat{\mathbf{W}}^{-T} \widehat{\mathbf{W}}^{+T} \begin{pmatrix} \mathbf{b}(0) & \mathbf{b}\left(\frac{1}{T'}\right) & \dots & \mathbf{b}\left(\frac{T'-1}{T'}\right) \end{pmatrix} \right\|_F^2 . \quad (5.85)
\end{aligned}$$

By use of the Moore-Penrose pseudoinverse, as discussed in section 5.3.2, the non-duplicated control points $[\mathbf{p}'] \widehat{\mathbf{W}}^{-T}$ that solve this problem are given by:

$$[\mathbf{p}'] \widehat{\mathbf{W}}^{-T} = \mathbf{X} (\widehat{\mathbf{W}}^{+T} \mathbf{B})^T (\widehat{\mathbf{W}}^{+T} \mathbf{B} (\widehat{\mathbf{W}}^{+T} \mathbf{B})^T)^{-1} . \quad (5.86)$$

Using this method to calculate maximum likelihood estimates of the trajectory vectors that explain each observed $(\mathbf{h}_{T_s}, \mathbf{g}_{T_s})$ gives a set of sample trajectory vectors that we can use to parameterise the B-spline distribution Ξ_Q . We make the same assumptions about this dis-

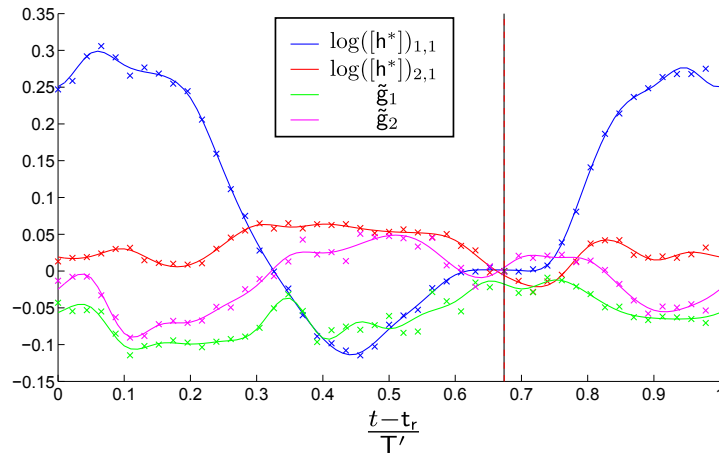


Figure 5.8: This figure shows an example of the result of fitting a B-spline to a quasi-periodic deformation vector subtrajectory. The crosses mark the subtrajectories of the first two principal components of the deformation mode model and of two of the matrix logarithm entries, and the solid lines of matching colours are the maximum likelihood B-splines that model them. The vertical red dashed line marks the time point $\frac{\check{t} - t_r}{T'}$, and the vertical black solid line behind it marks the candidate phase offset calculated over the B-spline using a method that we will describe in the next chapter. The data comes from a test based on square canonical grids over the 230108.FO1 data set that our results in the next section are partly based on. The parameters we used were $c = 23$, $d = 3$, $m = 40$.

tribution as we made about the distribution of the \mathbf{g} s in section 5.3.4, namely that it is an nc -dimensional multivariate Gaussian distribution for which the $nc - M$ principal components with the smallest eigenvalues capture pure noise, for some number M of principal deformation trajectory modes. The process by which we parameterise the distribution and calculate its modes is the usual linear PCA process, as described in that section.

Figure 5.8 shows the typical result of modelling a quasi-periodic deformation vector subtrajectory in this way. The candidate phase offsets of the maximum likelihood B-splines tend to be close to $\frac{\check{t} - t_r}{T'}$, as shown in the figure. Discarding the last $T - T'$ time steps occasionally has the unfortunate effect of discarding time point \check{t} , but this did not happen often enough to have a detrimental effect on our use of the candidate phase offsets, which we will discuss in the next chapter.

Another point that the figure highlights is that some of the matrix logarithm components of the deformation vectors were generally larger in magnitude than the components that came from the $\tilde{\mathbf{g}}_{T_s}$. As PCA is sensitive to the relative scales of the dimensions of the vectors that it is applied to, it seems that it would be appropriate to rescale the matrix logarithm components. But it is

not clear how we might go about identifying optimal scaling factors efficiently (a trial and error approach in which we test the effects of different scaling factors on the DTM’s reconstruction accuracy would be impractically inefficient). Nonetheless, we were able to achieve reasonable results without rescaling, as the next section will show.

5.5 Results

We shall now present some results that demonstrate the accuracy with which the procedures we have described for constructing deformation mode models and DTMs capture the deformations in two training sets from two different patients, and the extent to which they reduce the number of dimensions of the sequences of canonical grid deformations. Our intention is not to present a comprehensive study in which we construct general models that could be applied to any patient (we do not have enough videos in which we can manually track sufficiently dense landmark sets for that), but rather to provide evidence for the kinds of results we could expect from a larger study.

Table 5.4 provides a set of summary statistics for the two landmark sequences we used, and figures 5.11 and 5.12 show some frames from the sequences of deformation fields. The key thing to note is that there is a large change in the areas of the convex hulls of the landmarks for sequence 180608.HS, as shown in the table by the fact that the 25th area percentile is less than half the 100th percentile. This can also be seen directly in the figures of the deformation field sequences, and in the original video images shown in figure 5.3. Such large area changes did not occur in the 230108.FO1 sequence however, which meant that the h_{T_s} sequences of affine canonical grid registrations that we calculated were very different for each sequence.

We used four different quasi-circular grid representations in our deformation mode models and DTMs, which we shall refer to as “*Circular, Cartesian*”, “*Circular, Cartesian, Constrained*”, “*Circular, Polar*” and “*Circular, Polar, Unif. ang. scaling*”. The “*Cartesian*” grids used the Cartesian coordinate representation of the grids that we discussed in section 5.3.4. Of these, we carried out the registration to the canonical grid using the constrained affine registration

method from section 5.3.2 for those labelled “Constrained”, and for those not labelled as such, we used the unconstrained method. The grids labelled as “Polar” used the polar coordinate representation. We scaled the angular deviations of those labelled “Unif. ang. scaling” uniformly over all rings, and scaled the angular deviations of those not labelled as such according to the radial distances of the vertices from the origin. Using these four quasi-circular grid types allowed us to change each aspect of the representation in a controlled manner, so that we could see which aspects, if any, made a significant difference.

All of our tests were based on sets of 1000 similarity transformations of the canonical grids, randomly sampled as described in section 5.3.3. We will use the term *source frames* to refer to the initial frames that the similarity transformations were sampled with respect to (i.e. $t_r + \mathbf{o}$, for deformation field reference frame t_r and random phase offset \mathbf{o}), and *source grids* to refer to the states of the similarity-transformed canonical grids. To keep the tests as controlled as possible, we used one set of similarity transformations for each deformation field sequence over all four quasi-circular grid types. However, due to the square grids having a different boundary shape to the quasi-circular grids, the boundaries of the similarity transformation distributions for square grids differ to those of the quasi-circular grids, so we used separate sets of similarity transformations for them. We did however try to match the degrees of freedom in the square grids and the quasi-circular grids as closely as possible across all of our tests. The canonical grid parameters we used and the resulting degrees of freedom are given in table 5.1.

We defined the bounds on the isotropic scaling components of the similarity transformations by requiring the areas of all of the source grids to lie in the interval $[100^2, 180^2]$ pixels². As we explained in section 5.3.3, we ensured that the distribution of areas over this interval was approximately uniform by partitioning it into 20 subintervals and sampling $1000/20 = 50$ scaling factors within the bounds imposed by each subinterval. Table 5.2 gives the ranges of diameters and maximum edge lengths of the source grids imposed by this range of areas (by “maximum edge length”, we mean the maximum distance between neighbouring vertices in a grid). Comparing its “Max. Edge Length” column to the “Spc.” columns of table 5.4, we see that the resolutions of the canonical grids were much finer than the resolutions of the landmark sets – the distributions of the number of landmarks contained in the source grids are

Grid Type	Params	DoF
Square	$l = 10$	$2l^2 = 200$
Circular, Cartesian	$r = 6, s = 20$	$2((r - 1)s + 1) = 202$
Circular, Cartesian, Constrained	$r = 6, s = 20$	$2(r - 1)s = 200$
Circular, Polar	$r = 6, s = 20$	$2(r - 1)s = 200$
Circular, Polar, Unif. ang. scaling	$r = 6, s = 20$	$2(r - 1)s = 200$

Table 5.1: The parameters we used for each type of canonical grid and their degrees of freedom (DoF). The parameter symbols are the same as those we used in section 5.3.3. As mentioned earlier, the innermost rings of the unconstrained quasi-circular grids have just 2 DoF, because they are degenerate. However the innermost rings of the constrained quasi-circular grids (including the polar grids) have 0 DoF, as their vertices always lie at the origin.

Grid Type	Diameter Range	Max. Edge Length
Square	[141.4,254.6]	[11.1,20]
Circular	[113.8,204.8]	[11.4,20.5]

Table 5.2: This table shows the ranges of diameters and maximum edge lengths of the source grids. The maximum edge length of a grid is the maximum distance between neighbouring vertices. The diameters of the square grids are calculated as the distances between diagonally-opposite vertices. Their range of side lengths was [100, 180].

summarised in table 5.3. Comparing these maximum edge length ranges to the distributions of deformation field edge lengths shown in figure 5.9 however, it becomes clear that the source grids we used downsampled the deformation fields by factors of about 1-10. We had to do this for technical reasons related to the memory footprint of the data sets*, but nonetheless we still achieved reasonable results, as we shall discuss shortly.

*For each deformation mode model that we constructed, the memory footprint was over $1000DT$ doubles, where D is the number of dimensions of each canonical grid (D is $2l^2$ for the square grids and $2sr$ for the quasi-circular grids) and T is the length of the deformation field sequence. If we had chosen $l = 100$ for the square grids, then we would have had to store more than $8 \times 1000 \times (2 \times 100^2) \times 50B \approx 8GB$.

Seq.	Grid Type	Min. no. landmarks				
		25%	50%	75%	95%	100%
180608.HS	Circular	10	7	5	3	1
180608.HS	Square	8	5	4	2	0
230108.FO1	Circular	6	5	3	2	1
230108.FO1	Square	4	3	3	1	0

Table 5.3: This table shows the percentages of source grids that contained at least as many landmarks as are indicated in the table's body.

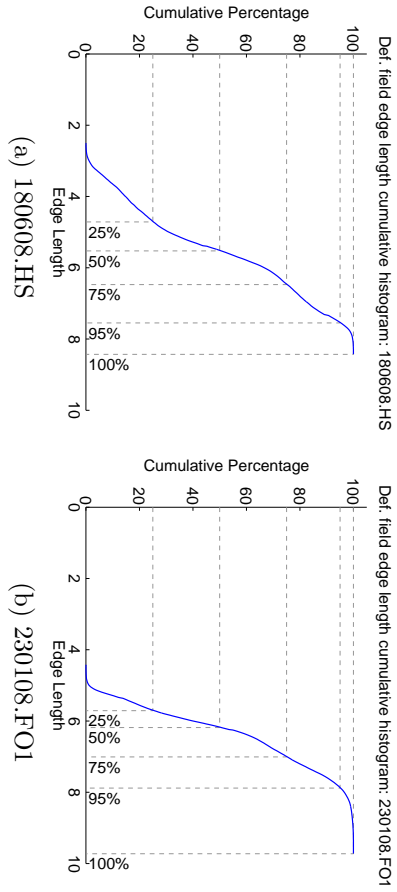


Figure 5.9: Cumulative distributions of deformation field edge lengths (in pixel units).

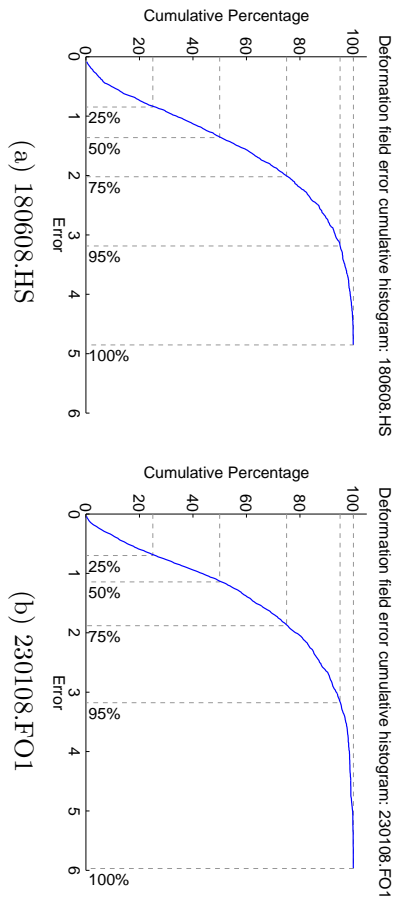


Figure 5.10: Cumulative distributions of deformation field landmark reconstruction errors (in pixel units).

Seq.	T	N	Area (p ²)			Dens. (100p) ⁻²			Diam. (p)			Spc. (p)			Disp. (p)		
			25	50	100	25	50	100	25	50	100	25	50	100	25	50	100
180608.HS	51	29	49k	81k	115k	2.8	3.6	6.6	(203, 353)	(275, 394)	(358, 449)	49	68	312	6	11	35
230108.FO1	59	30	81k	95k	109k	2.9	3.2	4.1	(335, 401)	(351, 450)	(368, 491)	53	72	333	5	7	45

Table 5.4: *Landmark sequences: summary statistics.* This table gives summary statistics for the manually-defined landmark sequences that the results in this section are based on, using the 25th, 50th and 100th percentiles of the indicated measurements, calculated over all frames. “ T ” denotes the number of frames that each sequence is defined over, “ N ” is the number of landmarks in each sequence, “Area” gives the areas of the convex hulls of the landmarks, “Dens.” is the mean number of landmarks per 10000 square pixels in each frame, “Diam.” refers to the differences between the most positive and most negative projections of the landmarks onto their principal components (smallest principal component first), “Spc.” gives the distances between connected landmarks in their Delaunay triangulations and “Disp.” gives the distances landmarks move in-between consecutive frames. The unit “p” stands for “pixel”.

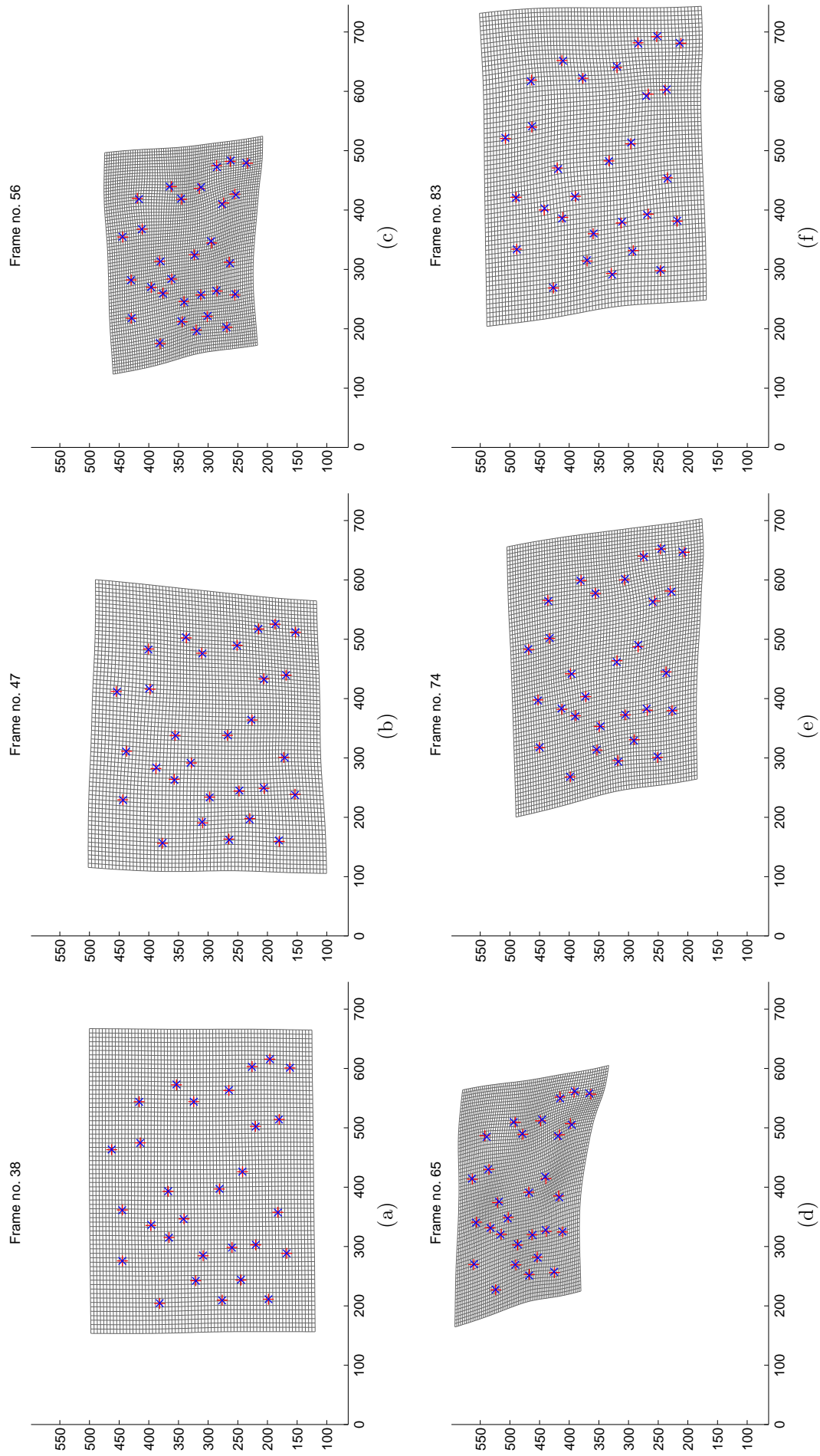


Figure 5.11: Six deformation fields for video 180608.HS. The red pluses show the true landmark positions, and the blue crosses show the deformation fields' estimates of their positions.

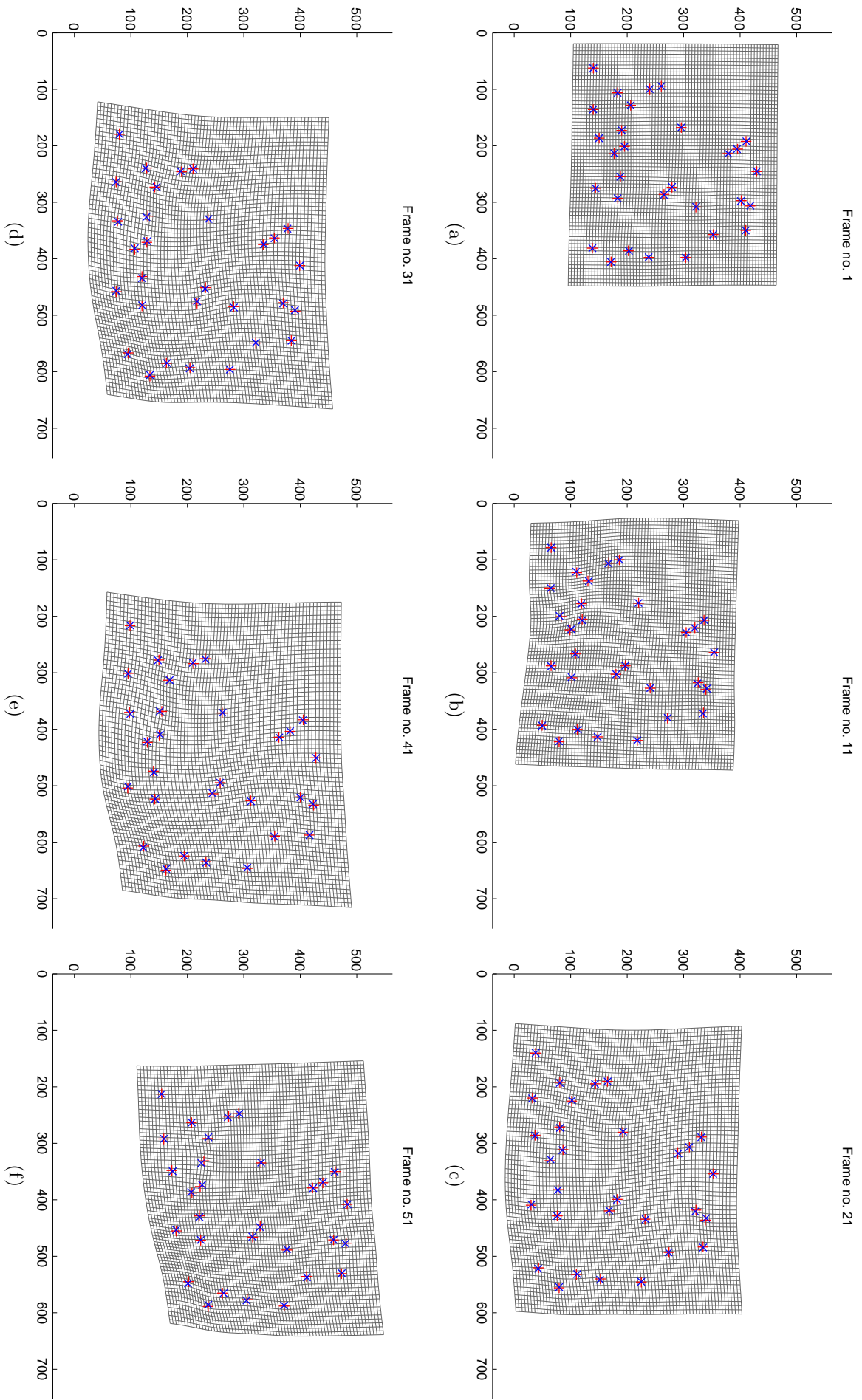


Figure 5.12: Six deformation grids for video 230108.FO1. As before, the red pluses show the true landmark positions, and the blue crosses show the deformation fields' estimates of their positions.

Our tests involved constructing deformation mode models and DTMs from the similarity transformations, using them to reconstruct the positions of the original landmarks for varying numbers of modes and B-spline control points, measuring the errors in these reconstructions and using the errors to determine the numbers of modes and control points we should select for later applications of the models. The landmark reconstruction errors never reach zero, due to the errors in the deformation fields, but they tell us the minimum numbers of modes and control points that the models need to achieve tolerable degrees of error. They are more useful measures of model fit than the errors with which the models reconstruct regions of the deformation fields, which do approach zero when the number of control points and the canonical grid resolution are high enough, but give no indication as to whether or not overfitting is occurring (i.e. whether or not the models are beginning to fit the deformation field errors rather than the true myocardial deformations that the fields approximate).

We will now describe each test in more detail and give an analysis of the results.

5.5.1 Deformation Mode Models

For each sampled random phase offset \mathbf{o} , random similarity transformation \mathbf{f} and set of deformation field frame numbers $T_s = \{t_r, \dots, t_r + T - 1\}$, we used the following process to reconstruct the trajectory \mathbf{y}_{T_s} of each landmark \mathbf{y} that lay in the source grid in frame $t_r + \mathbf{o}$ using m of the k principal components of each deformation mode model:

1. Determine the quadrilateral $\square\mathbf{abcd}$ of $\mathbf{f}(\mathbf{g})$ that contains $\mathbf{y}_{t_r+\mathbf{o}}$ and the bilinear interpolation coefficients α and β which, when linearly combined with the quadrilateral's vertices, give the landmark's position exactly. We use the methods we described in section 5.3.1 for this.
2. Calculate a reconstruction \tilde{g}_{T_s} of the sequence of non-affine deformed canonical grids g_{T_s} by projecting them onto the model's principal components, discarding the $k - m$ principal components with the smallest eigenvalues, and projecting them back into the canonical grid's coordinate system (in Cartesian coordinates).

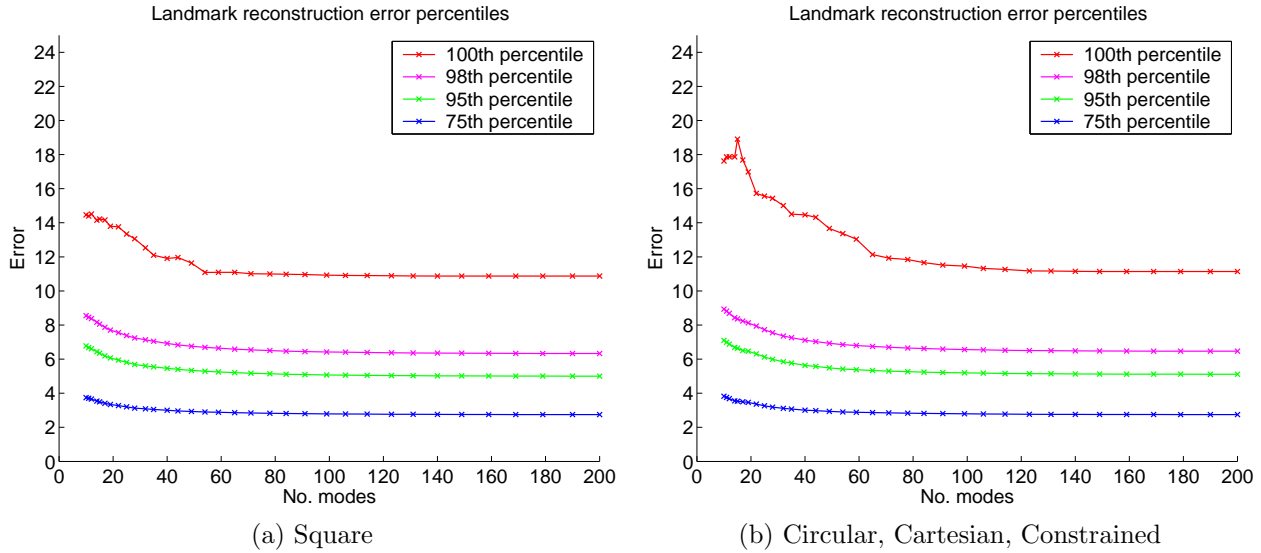


Figure 5.13: Landmark reconstruction error percentiles for two deformation mode models trained and tested on 230108.FO1.

3. Use \tilde{g}_{T_s} , α , β and $\square abcd$ to calculate a reconstruction \mathbf{y}'_{T_s} of \mathbf{y}_{T_s} 's trajectory in the canonical grid's coordinate system.
4. Transform \mathbf{y}'_{T_s} back into the coordinate system of the deformation fields. As the deformation mode model does not store affine transformation information, this has to be done with the inverted affine registration functions $h_{T_s}^{-1}$.

We did this for each canonical grid representation over a set of 34 values of m , approximately quadratically distributed over $[10, 200]$ (quadratically distributed so that $[10, 200]$ was most densely sampled at the low end, where small changes in m make the greatest difference to each model's accuracy). We constructed each model using the deformation data from just one video at a time, but we calculated the landmark reconstruction errors over both videos, for cross-validation. The first 10 modes of the models for 230108.FO1 are shown in figure 5.15; the modes for 180608.HS are similar.

Figure 5.13 shows how the reconstruction error percentiles varied with m for two deformation mode models that we trained and tested on sequence 230108.FO1. We have summarised the variations of the 50th, 75th, 95th, 98th and 100th error percentiles with m for all four pairs of training sequences and test sequences in figure 5.14. In that figure, the error bars denote

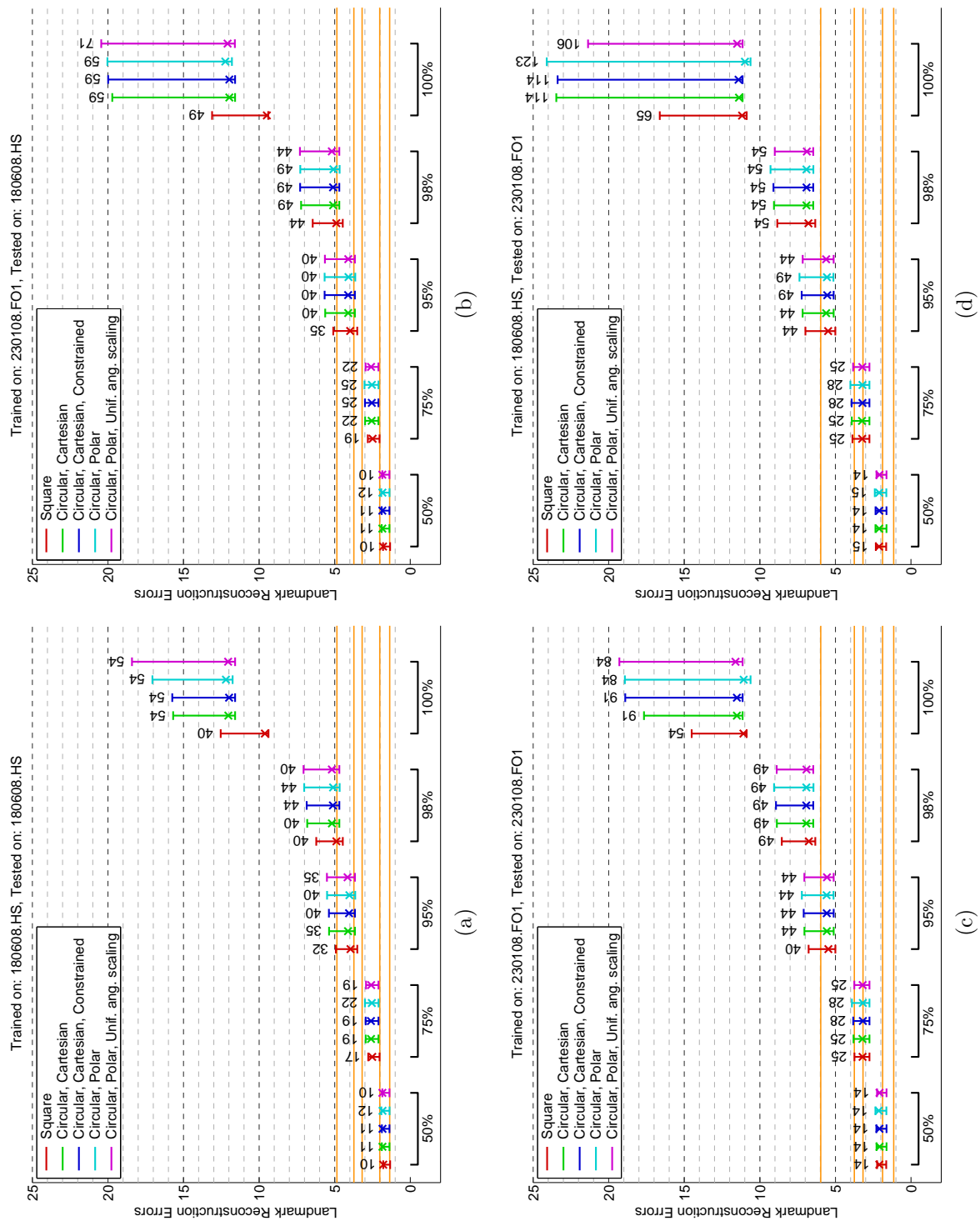


Figure 5.14: Deformation mode model landmark reconstruction error percentile ranges (in pixel units). See text for description.

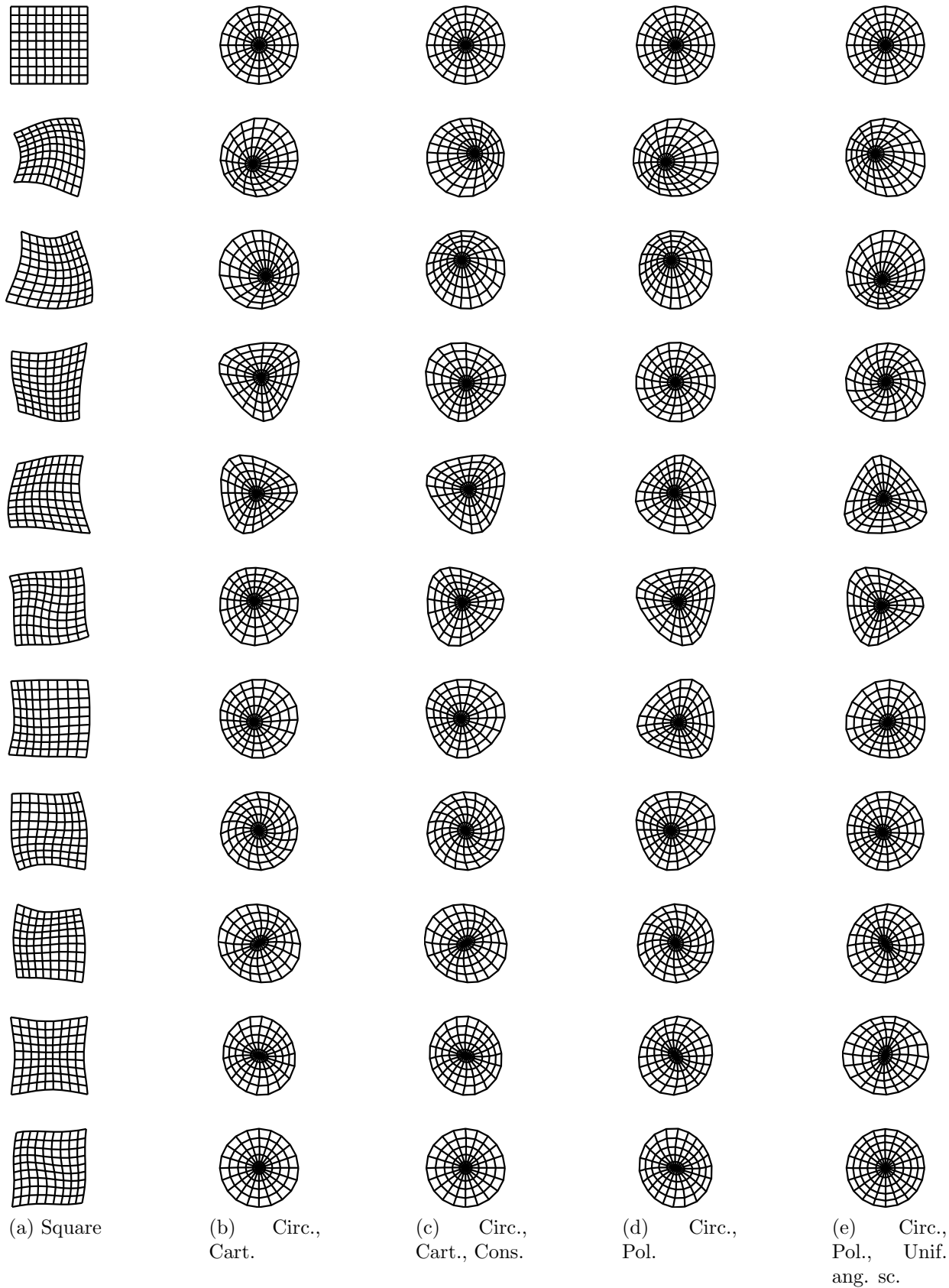


Figure 5.15: These figures show the mean and first 10 modes of the deformation mode models for all 5 grid types. The top row contains the means, and the remaining rows show the modes, with eigenvalues decreasing from top to bottom. To highlight the kinds of deformations that the modes represent, we have scaled them all by nine standard deviations.

the upper and lower bounds on the ranges of the percentiles, as functions of m . The numbers above them are the smallest values of m at which the percentiles fall to within 0.5 pixels of the minima. The crosses mark the values of the percentiles at these thresholds. Finally, to facilitate comparison with the deformation field landmark reconstruction errors, we have marked the same percentiles of those errors with orange horizontal lines.

In all cases, values of m in the range $[32, 49]$ were sufficient to bring 95% of the errors to within the range of errors of the deformation fields. In the 180608.HS video, for $m \geq 25$, the 50th and 75th percentiles agreed with those of the deformation fields. Also, the five grid types perform very similarly up to the 98th percentile; the only significant difference between them seems to be that all of the circular grids produce much larger outliers at the 100th percentile. The cross-validation tests suggest that the models generalise well, in that up to the 98th percentile, the percentile ranges for each test sequence are almost identical across the two training sequences.

5.5.2 Deformation Trajectory Mode Models

Under the results of the previous section, we constructed all of our DTMs using 40-mode nested deformation mode models. Although some of the deformation mode models required up to 49 modes for 95% of the errors to lie within the error range of the deformation fields, the 95th error percentiles with 40 modes were still within the 7 pixel limit that we were willing to accept.

Reusing the previous section's notation, we used the following procedure to reconstruct the trajectory $\mathbf{y}_{T'_s}$ of each landmark \mathbf{y} in each source grid $\mathbf{f}(\mathbf{g})$ with M of each DTM's K modes, where $T'_s = [t_r, \dots, t_r + T' - 1]$, and T' is the length of the source grid's quasi-periodic subtrajectory, calculated as in section 5.4.3:

1. As before, determine the quadrilateral $\square abcd$ of $\mathbf{f}(\mathbf{g})$ that contains $\mathbf{y}_{t_r+\circ}$ and the bilinear interpolation coefficients α and β .
2. For each pair (h_t, g_t) of affine and non-affine transformations, calculate the 44-dimensional deformation vector $\mathbf{x}_t = \Upsilon(h_t^{*-1} \circ h_{t_r+\circ'}^*, g_t)$ as described in section 5.4.3, where h_t^* is the

non-translational part of h_t and o' minimises $g_{t_r+o'}$'s projection onto the deformation mode model's principal components.

3. Calculate the trajectory vector \mathbf{p} of the periodic B-spline that best fits the deformation vector sequence.
4. Project \mathbf{p} onto the DTM's first M modes and reconstruct it to give a trajectory vector $\tilde{\mathbf{p}}$.
5. For each $t \in T'_s$ calculate a reconstruction $\tilde{\mathbf{x}}_t$ of \mathbf{x}_t by evaluating the B-spline: $\tilde{\mathbf{x}}_t = \mathbf{q}(\frac{t-t_r}{T'}; \tilde{\mathbf{p}})$.
6. For each $t \in T'_s$, reconstruct $(h_t^{*-1} \circ h_{t_r+o'}^*, g_t)$ by evaluating $(\tilde{h}_t^{*-1}, \tilde{g}_t) = \Upsilon^+(\tilde{\mathbf{x}}_t)$.
7. Transform \tilde{h}_t^{*-1} into a transformation \hat{h}_t^{-1} that maps the canonical grid coordinate system to the deformation field coordinate system, by setting \hat{h}_t^{-1} 's non-translational component to $\tilde{h}_t^{*-1} \circ h_{t_r+o'}^{*-1}$, and its translational component to that of h_t^{-1} .
8. Use $\tilde{g}_{T'_s}$, α , β and $\square \mathbf{abcd}$ to calculate a reconstruction $\tilde{\mathbf{y}}_{T'_s}$ of landmark \mathbf{y} 's trajectory in the canonical grid's coordinate system by bilinear interpolation.
9. Transform $\tilde{\mathbf{y}}_{T'_s}$ back into the coordinate system of the deformation fields with the reconstructed inverse affine registration functions $\hat{h}_{T'_s}^{-1}$.

We did this for all values of $M \in \{30, 40, \dots, 80, 90\}$ using degree 3 B-splines with c control points, for all $c \in \{13, 18, \dots, 33, 38\}$, so that the B-splines had 10, 15, \dots , 30, 35 degrees of freedom. As the two deformation sequences were so different, it would not have been possible to achieve good cross-validation results with such small values of M , so we only calculated the landmark reconstruction errors for the sequences that we trained the DTMs on.

Figures 5.16 and 5.17 summarise the 50th, 75th, 95th, 98th and 100th error percentiles much as before, except that this time the error bars show the positions of each percentile. The 50th to 98th percentiles were very similar for all grid types in both test sequences. The greatest differences in 230108.FO1 occurred with $c = 13$, $M = 30$, where the 95th and 98th percentiles of the “Circular, Cartesian, Constrained” and “Circular, Polar” grid types were almost 1 pixel

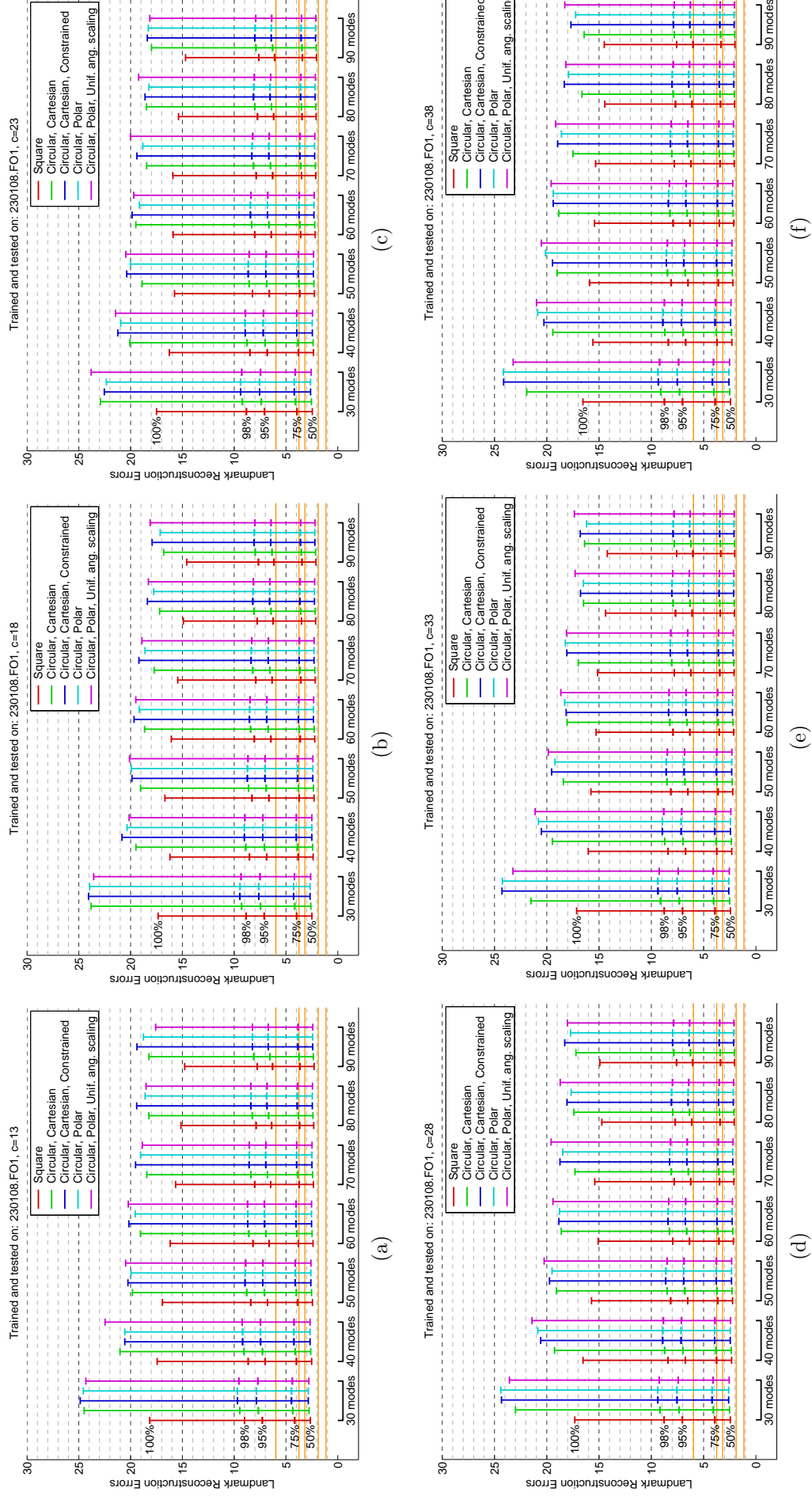


Figure 5.16: DTM landmark reconstruction error percentiles for 230108.FO1 under different numbers of B-spline control points (in pixel units). See text for description.

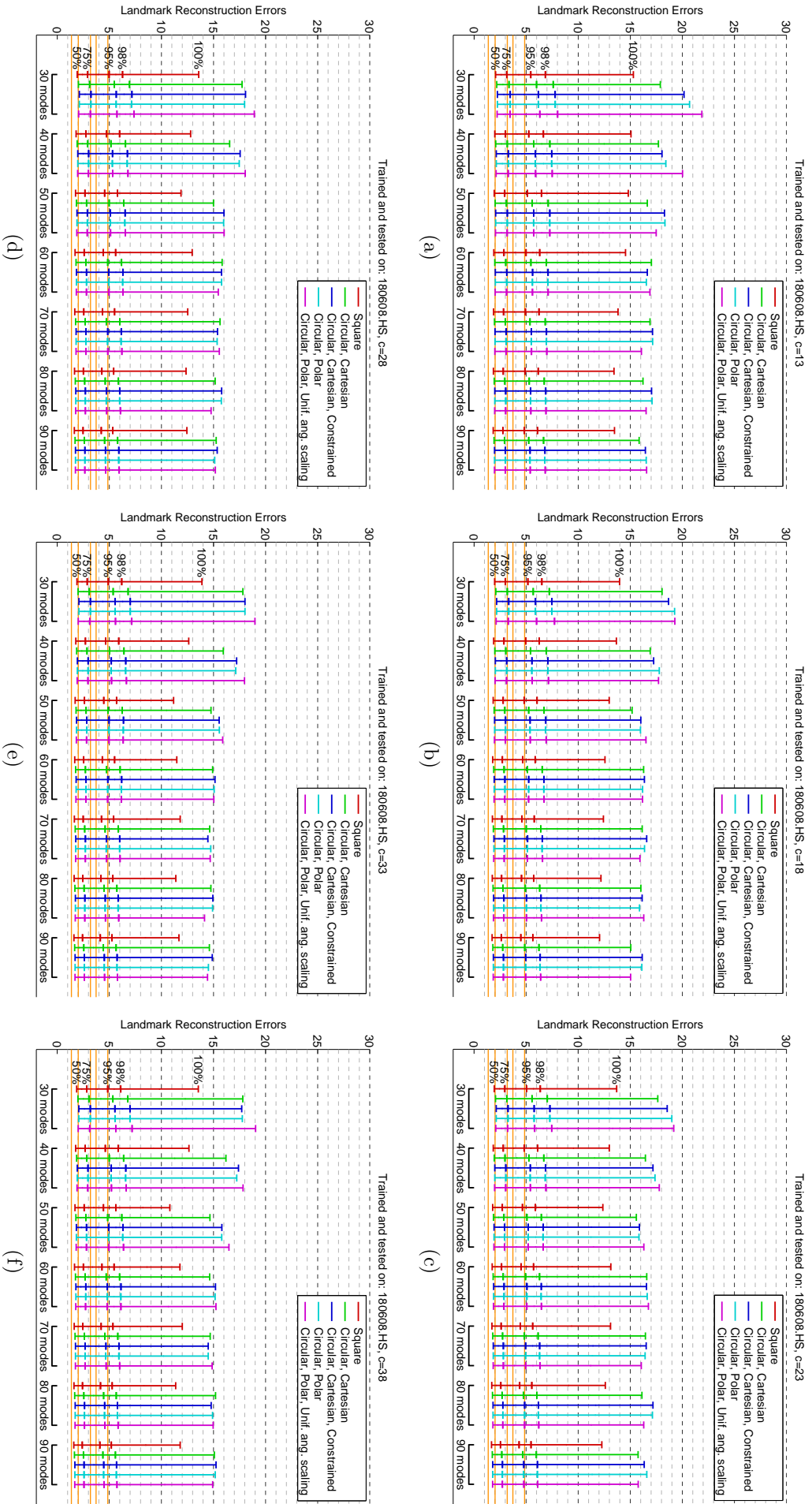
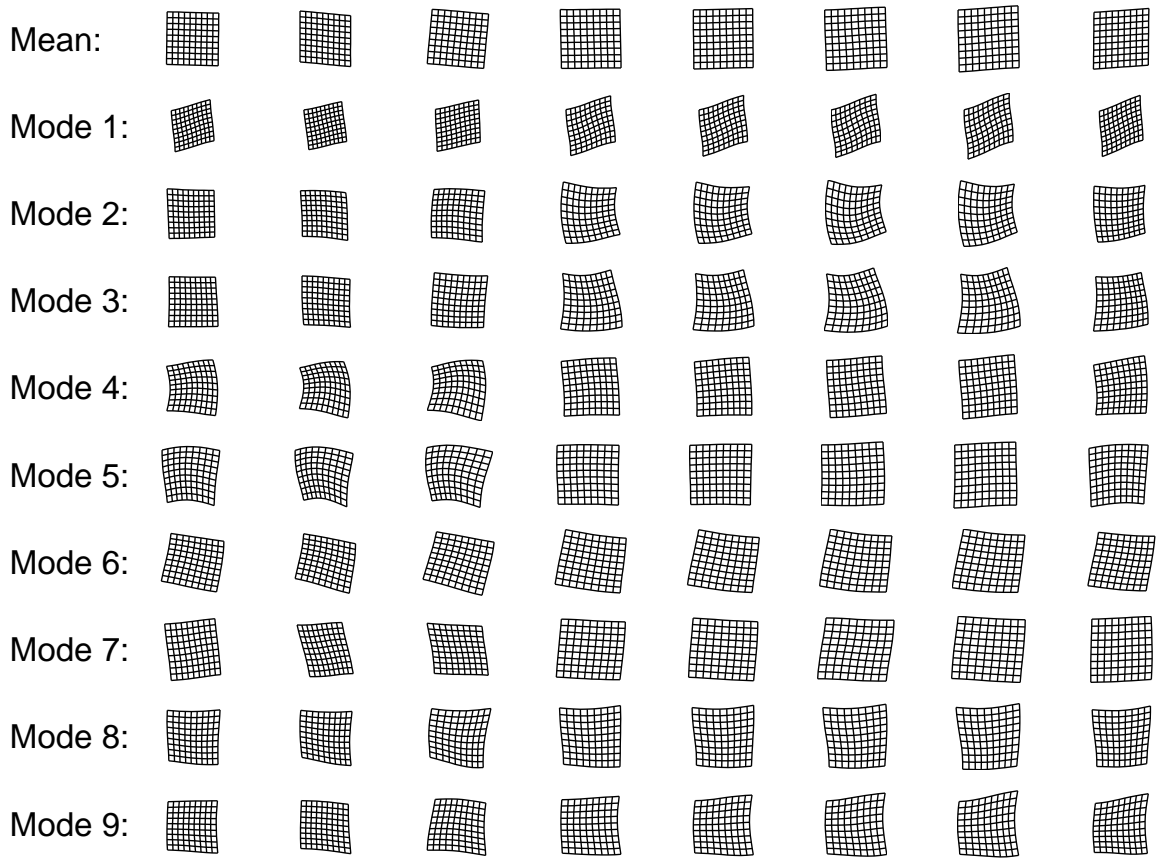
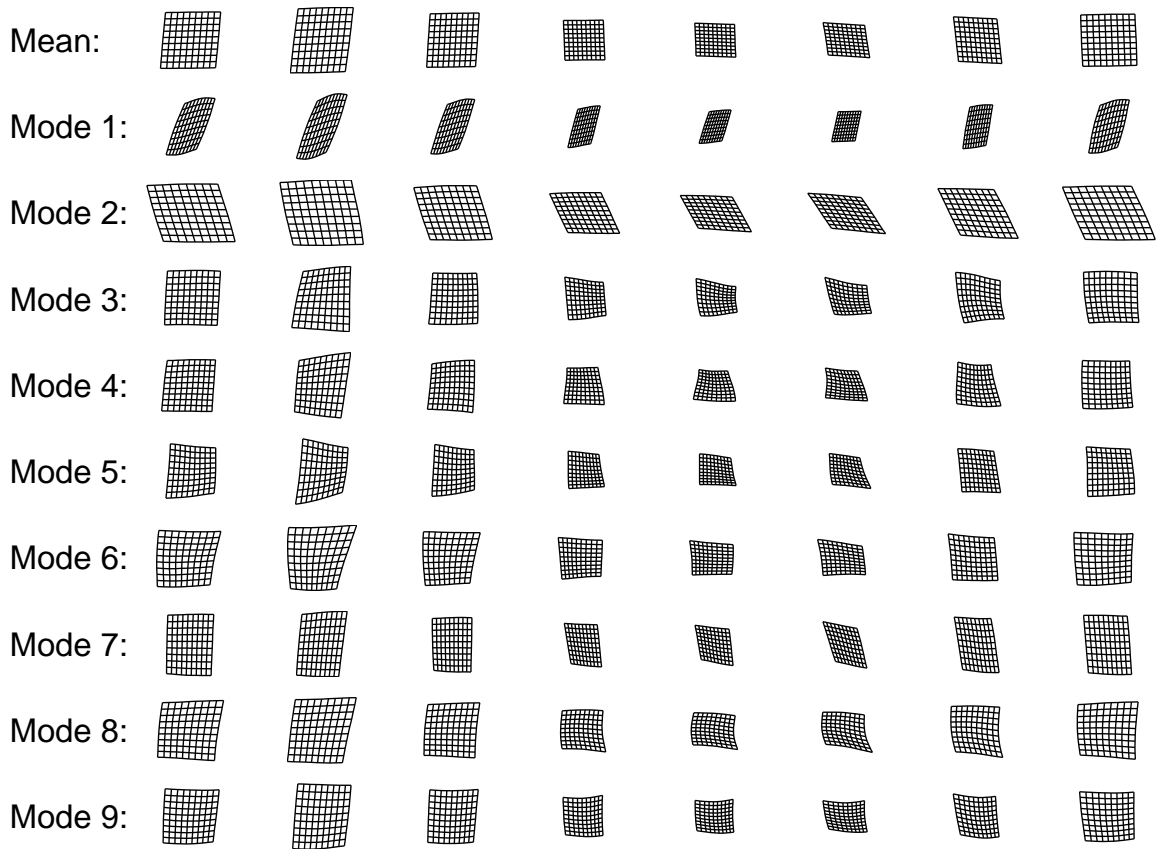


Figure 5.17: DTM landmark reconstruction error percentiles for 180608.HS under different numbers of B-spline control points (in pixel units). See text for description.



(a) 230108.FO1



(b) 180608.HS

Figure 5.18: Eight samples from left to right of the mean trajectories and first nine modes of the square grid DTMs. To highlight the kinds of deformation sequence that the modes represent, we have scaled them by six standard deviations.

c	M (Sq.)	M (Circ., Cart.)
13	30	60
18	30	40
23	30	40
28	30	30
33	30	30
38	30	30

Table 5.5: Sampled values of M for which the 98th percentiles of the “Square” and “Circular, Cartesian” landmark reconstruction errors fell within the 7 pixel limit for 180608.HS.

c	M (Sq.)	M (Circ., Cart.)
13	50	60
18	40	50
23	40	50
28	40	50
33	40	40
38	40	40

Table 5.6: Sampled values of M for which the 95th percentiles of the “Square” and “Circular, Cartesian” landmark reconstruction errors fell within the 7 pixel limit for 230108.FO1.

greater than those of the “Square” grid type. These minor differences diminished with increasing c and M . Similar behaviour occurred in 180608.HS where, for $M = 30$, the 98th percentiles for the “Circular, Polar, Unif. ang, scaling” grid type were 1-2 pixels greater than those of the “Square” grid type, but again these differences diminished with increasing M .

The 50th and 75th percentiles for all grid types were up to 2 pixels greater than those of the deformation fields in 230108.FO1, and at most a little over 1 pixel more in 180608.HS. The differences between the 95th and 98th percentiles of the grids and the deformation fields were greater, but for sufficiently large values of M , the 95th percentiles fell within the acceptable 7 pixel limit for 230108.FO1 and the 98th percentiles fell within that limit for 180608.HS. The sampled values of M for which this happened with the “Square” and “Circular, Cartesian” grid types (which, of all the quasi-circular grid types, had the lowest percentiles) are summarised in tables 5.6 and 5.5. From these tables and the graphs, it seems that the DTMs based on square canonical grids consistently lead to the simplest representations of the deformation sequences, since for each value of c , they minimised the sampled value of M for which the 95th percentile of the errors was acceptably low, and they also minimised the magnitudes of the 100th percentile outliers.

Although the B-splines are only intermediate objects in the reconstruction process, it is nonetheless preferable to minimise c , so as to avoid overfitting during the model construction process. So, using square grid DTMs with $c = 23$ and $M = 50$ for both sequences seems to be a rea-

sonable compromise between the conflicting desires of minimising c , M and the reconstruction errors. The 50th to 98th percentiles only decreased by up to 0.5 pixels for the values of $c > 23$ or $M > 50$ that we tested in 230108.FO1, and by up to 1 pixel in 180608.HS. Figure 5.18 shows samples from the first nine modes of these models.

5.6 Conclusion

In this chapter, we have discussed the methods that we use to construct statistical Deformation Trajectory Models (DTMs) of the sequences of 2D deformations that the myocardium undergoes. This allows us to sample empirically-justified periodic sequences of deformations that we can use to estimate the parameters of our particle filter’s likelihood functions.

We begin the DTM construction process by fitting deformation fields to sequences of manually-placed landmarks, and using these to calculate a PCA model that we call a “deformation mode model”, which gives the principal non-affine deformation modes of a canonical subgrid randomly selected from somewhere within the image of the deformation fields. We carried out cross-validation tests on these models, and the results seemed to suggest that a single video may provide sufficient information with which to construct deformation mode models that perform well on new patients.

The final stages of the DTM construction process involve concatenating the affine transformation parameters to the deformation modes discovered by the deformation mode model to form “deformation vectors”, fitting periodic B-splines to sequences of deformation vectors that describe the deformation of a canonical grid over a cardiac cycle, and then using PCA again to calculate the principal modes of the B-spline control points. Our tests suggested that using square grids (with vertices specified in Cartesian coordinates) to construct DTMs seems to lead to the simplest representation of the sequences of deformations. It was not possible to carry out cross-validation tests with the DTMs, as it was clear that the sequences of affine deformations were very different in the two videos we used. So to construct a general-purpose DTM using 2D coordinates, we will have to combine deformation information from a large number of patients.

Chapter 6

Towards a Strategy for Coping with Particle Loss

6.1 Synopsis

In this chapter, we shall present our initial attempts at answering the three questions that we posed on page 115. Our answer to the first question, concerning the selection of the likelihood function parameters, makes use of the DTMs we developed in the previous chapter. More specifically, we sample sequences of plausible deformations from the DTMs, and use them to warp the regions surrounding the initial patch locations, so that we can calculate sample sets of the differences that we may expect to observe between matching patches/subregions as a result of myocardial deformations. We then use these sample sets to calculate maximum-likelihood estimates of the likelihood function parameters.

In answer to the second question, about detecting when particles are lost, we employ a likelihood-ratio test that compares the likelihood of a patch's appearance under the hypothesis that it is lost to the likelihood of its appearance under the hypothesis that it is not lost. We use the patch likelihood function L_t to define the latter likelihood, and a *background likelihood* histogram to define the former. As with L_t 's parameters, we construct this histogram using DTM-based simulated deformations, except that this time we calculate differences between randomly se-

lected patches/subregions that do not match the reference patch/subregion. We select the test threshold by minimising the probability of a misclassification under L_t and the background likelihood function.

For the third question, on the restoration of lost particles, we employ a simple method based on sampling from the particles that have been labelled “not lost”. This method works well enough when only a few particles are lost, but cannot be used when all particles are labelled “lost”. We discuss some alternatives, such as using information from the first frame, and we describe some of their limitations.

We then end the chapter with an evaluation of our particle loss test’s performance and an evaluation of the performance of our particle filtering algorithm under all of the methods that we have described in this thesis.

6.2 Introduction

The success of any particle filtering algorithm rests on the fidelity of its representation of the posterior distribution of the hidden states. As explained in [61] and [29], the variance of the particle weights tends to increase with time during sequential importance sampling, the eventual effect of which is usually that the weights degenerate to a state in which all but one of the particles have weights close to zero (one particle will have a large weight in this scenario as a result of normalisation, but it will not necessarily be located anywhere near a posterior mode).

The most common way of dealing with this degeneration problem involves calculating the effective sample size (ESS), first proposed in [60] and defined with respect to the weights $w_t^{[i]}$ of the n time t particles as

$$\frac{1}{\sum_{i=1}^n \left(w_t^{[i]}\right)^2} . \quad (6.1)$$

The ESS’s value varies between 1, when all but one of the weights are 0, and n , when all of the weights are $\frac{1}{n}$.

Suppose we have two estimates \bar{f}_{m,π_t} and \bar{f}_{n,ℓ_t} of the posterior expectation of an arbitrary function $f(\Theta_{1:t})$, the former being calculated by a Monte Carlo integration process in which m samples are drawn directly from the posterior, and the latter being calculated by drawing n samples from the importance sampler ℓ_t . Kong explains in [60] that the expression $n \sum_{i=1}^n \left(w_t^{[i]}\right)^2$ is an approximation of the relative efficiency of \bar{f}_{m,π_t} and \bar{f}_{n,ℓ_t} , i.e. if \bar{f}_{n,ℓ_t} has x times the variance of \bar{f}_{m,π_t} , the value of this expression should be close to x . So given the well-known Monte Carlo integration theory result that the variance of \bar{f}_{m,π_t} is proportional to $\frac{1}{m}$ ([47]), it follows that the ESS approximates the number m of samples for which \bar{f}_{m,π_t} has the same variance as \bar{f}_{n,ℓ_t} .

A typical solution to the degeneration problem is to resample the particle set whenever the ESS falls below a certain threshold, and then reset all of the particle weights to $\frac{1}{n}$. The resampling step involves sampling n times with replacement from the old particle set, selecting particle state $\Theta_{1:t}^{[i]}$ with probability $w_t^{[i]}$ each time. This may be an effective solution when n is large and the ESS decreases slowly with time. But if the ESS drops suddenly, the particle set may degenerate before the resampling step occurs, so that the particle set only has a small number of distinct states after resampling (it is said to be *impoverished* when this happens).

A common event that causes the ESS to drop suddenly is a period of occlusion (due, for example, to specular highlights not detected by our mask $\mathcal{F}_{O,t}$, or due to $\mathcal{F}_{O,t}$ masking out very large regions) during which the particles disperse. Such events are particularly problematic for our importance sampler ℓ_t when the time $t-1$ image \mathbf{Z}_{t-1} is used to describe the appearance of the subregions within patch $\mathbf{V}(\Theta_{t-1}^{[i]})$, especially for small n and occlusion periods lasting for more than a couple of frames, as the particles are unlikely to remain close enough to the true state for the pixels within the patch boundaries they define to be a reliable source of information about the appearance of the patch at a later time when the occlusion has passed. In fact, this misdirection induced by the persistent use of the previous image is the principal mechanism through which errors accumulate and particle sets degenerate in our application.

In addition, when using small n , it is crucial to maximise both the number of particles that are in high-posterior-probability states and the diversity of these particles, so that during periods of genuine ambiguity (such as when a formerly bent vessel momentarily straightens), the particle

filter may maintain a representation of as wide a range of probable patch state hypotheses as possible. Given this requirement and the misdirection problem of the previous paragraph, we propose an alternative solution to the problem of maintaining a faithful representation of the posterior in which, after the generation of each new particle set, we assess each particle individually to determine whether or not it is “lost” (i.e. in a state of unusually low probability), and take active steps to restore lost particles to high-probability states in the next frame.

The task of identifying when a particle is lost is best suited to the particle filter component that evaluates the (dis)similarity of each sampled particle state to its expected appearance (i.e. the component that evaluates the Δ_L function that we introduced in chapter 4). By using reliable information about what the patch and the region surrounding it should look like at time t , this component can carry out statistical hypothesis tests to infer whether or not each particle is lost.

In section 6.4, we will consider how these tests may be carried out under information derived from the first frame only by using the DTMs we developed in the previous chapter to simulate the appearance of the patch and its background in later frames. These simulations also provide us with a way of parameterising both the local likelihood functions that define the importance sampler and the patch likelihood functions L_t . So we will cover their parameterisation in that section, thus providing answers to the first two questions that we posed on page 115. Of course, simulations based on the first frame alone can only provide an initial estimate of the appearance changes that the myocardium undergoes, so an interesting topic for future research is how we might go about improving the reliability of these tests by incorporating information from later frames, in which we are less certain about the true state of the patch.

After discussing our particle loss tests, we will consider some methods for restoring lost particles, and then we will conclude with an analysis of the accuracy of our particle filter’s performance over the two video sequences that we have used throughout this thesis.

6.3 Detecting Particle Loss

Before we can detect and restore lost particles, it is important for us to make the condition under which we consider them to be lost precise. Returning to the notation we introduced in sections 3.2.1 and 3.3, the theoretically ideal way of defining whether or not the i^{th} particle is lost at time t involves a consideration of the marginal posterior probability density of the event $\Theta_t = \Theta_t^{[i]} \mid \Theta_t^{[i]}$, where Θ_t is the unobservable time t patch state, and $\Theta_t^{[i]}$ is the observed state of particle i at time t . The marginal posterior distribution π'_t is defined with respect to the full posterior π_t by marginalising out the subtrajectory $\Theta_{1:t-1}$:

$$\pi'_t(\Theta_t) \triangleq \int \pi_t(\Theta_t, \theta_{1:t-1}) d\theta_{1:t-1} . \quad (6.2)$$

This distribution captures all of the information we have about the patch's state at time t . So we consider the particle to be lost at time t if and only if it lies in a region of this distribution corresponding to states that the patch has an unusually low marginal posterior probability of being in. The following definitions make this more precise:

Definition 6.1. *Let Ω'_t be the region of Θ_t 's domain Ω defined by the following properties:*

1. *under π'_t , the event $\Theta_t \in \Omega'_t$ occurs with some small, prespecified probability p ;*
2. *for all $\theta \in \Omega - \Omega'_t$ and $\theta' \in \Omega'_t$, $\pi'_t(\theta) > \pi'_t(\theta')^*$.*

Definition 6.2. *A particle in state θ at time t is **lost** if and only if $\theta \in \Omega'_t$.*

As a simple example of these definitions, if π'_t was a univariate Gaussian distribution and we set $p = 1\%$, Ω'_t would be the union of the real numbers less than the 0.5th percentile and those greater than the 99.5th percentile.

A good value of p to choose would be one such that if all of the time t particle states were truly sampled from π'_t , the expected number contained in Ω'_t would be $\ll 1$. I.e., for n particles, p should satisfy $pn \ll 1$.

*Note that this property may not be satisfiable for some values of p if there are regions of π'_t with non-zero Lebesgue measure over which π'_t is uniform. The most extreme example of this is when π'_t is uniform everywhere over Ω , in which case this property is unsatisfiable for all $p < 1$.

6.3.1 Likelihood-Ratio-Based Particle Loss Inference

Needless to say, it would be impractical to attempt to carry out particle loss tests based on the definitions above, due, not only to the fact that the integrals that define π'_t (which includes the normalising constant of eq. (3.3)) are intractable to evaluate, but also to the difficulties we would face in trying to determine Ω'_t 's boundary, even if we could evaluate π'_t . Recalling eq. (3.9), it may at first seem reasonable to approximate $\pi'_t(\theta_t)$ using the particle weights $w_t^{[i]}$:

$$\pi'_t(\theta_t) \approx \sum_{i=1}^n \delta_{\theta_t, \Theta_t^{[i]}} w_t^{[i]} . \quad (6.3)$$

In cases where this is a good approximation of the value of π'_t at the particle states, we could try sorting the particles in ascending weight order, accumulating their weights, and considering the first m of these sorted particles to lie in Ω'_t if the sum of their weights is $< p$. This method would be unreliable if a large number of particles were to drift into Ω'_t at around the same time however, as might happen if the particles dispersed as a result of occlusions, so an alternative that assesses each particle individually would be preferable.

By eq. (6.2), the expressions we gave for the full posterior and its terms on page 44, and the expressions we gave for the likelihood $\Lambda_t(\theta_t)$ and its terms on page 51, π'_t satisfies

$$\begin{aligned} \pi'_t(\theta_t) &\propto L_t(\theta_t; \Theta_0, \mathbf{Z}_{\mathbf{L}, [0, t]}) H_t(\theta_t) \\ H_t(\theta_t) &\triangleq \int \ell_t(\theta_t; \theta_{0:t-1}, \mathbf{Z}_{\ell, 0:t}) p_i(\theta_t) \pi_{t-1}(\theta_{0:t-1}) d\theta_{1:t-1} . \end{aligned} \quad (6.4)$$

As we take the prior $p_i(\theta_t)$ to be uniform, $H_t(\theta_t)$ is proportional to the expected value of $\ell_t(\Theta_t)$ under the time $t - 1$ posterior π_{t-1} , i.e.

$$H_t(\theta_t) \propto E_{\pi_{t-1}}[\ell_t(\theta_t; \Theta_{0:t-1}, \mathbf{Z}_{\ell, 0:t}) \mid \Theta_0, \mathbf{Z}_{\ell, 0:t}] . \quad (6.5)$$

Viewing L_t as an expert that turns our Product-of-Experts model of the importance sampler into a Product-of-Experts model of the full likelihood function $\Lambda_t(\theta_t)$, the role of ℓ_t should be to reinforce, and not undermine, the information that L_t provides. More specifically, for all θ_t

and θ'_t , L_t and ℓ_t should ideally satisfy

$$\pi'_t(\theta_t) \geq \pi'_t(\theta'_t) \Rightarrow L_t(\theta_t; \Theta_0, Z_{L,[0,t]}) \geq L_t(\theta'_t; \Theta_0, Z_{L,[0,t]}) . \quad (6.6)$$

In cases where the initial patch is highly distinguishable from its surroundings, the non-translational deformations between consecutive frames are not large, and the non-affine deformations between time 0 and time t' are not large for each $t' \in \{1, \dots, t\}$, each importance sampling distribution $\ell_{t'}(\theta_{t'}; \Theta_{0:t'-1}, Z_{\ell,0:t'})$ should be in close agreement with $L_{t'}(\theta_{t'}; \Theta_0, Z_{L,[0,t']})$ on the probabilistic ordering of values of $\theta_{t'}$ because:

- a) the patch's distinguishability implies that $L_{t'}$ is a unimodal function of $\theta_{t'}$;
- b) for each pair $(S_{t'-1}, S_{t'})$ of square subregions such that $S_{t'-1}$'s centre lies within the time $t'-1$ patch $\mathbf{V}(\Theta_{t'-1})$ and $S_{t'}$'s centre lies at the corresponding time t' position, the regions of images $Z_{t'-1}$ and $Z_{t'}$ bounded by $S_{t'-1}$ and $S_{t'}$ should look similar to each other, since the non-translational deformations between consecutive frames are small;
- c) some $S_{t'}$ s should look different to all other time t' subregions that are involved in the definition of $\ell_{t'}(\theta_{t'}; \Theta_{0:t'-1}, Z_{\ell,0:t'})$, which again follows from the patch's distinguishability;
- d) each time t' patch $\mathbf{V}(\Theta_{t'})$ should look similar to an affine transformation of the initial patch state $\mathbf{V}(\Theta_0)$, since the non-affine deformations are not large.

If the subregion grid centres are sufficiently densely distributed, the distinguishability of the patch together with our results from chapter 3 on exponential likelihood ratio growth suggests that the importance sampling distributions $\ell_{t'}$ should be close to delta functions centred at the true patch states. In turn, this, together with the expected agreement between $L_{t'}$ and $\ell_{t'}$, suggests that the time $t-1$ posterior π_{t-1} should be close to a delta function centred at the true patch subtrajectory $\Theta_{0:t-1}$ (since, under a uniform prior, the posterior is proportional to the product of the patch likelihoods and the importance sampling distributions). Hence by eq. (6.5), it follows that

$$H_t(\theta_t) \approx k \ell_t(\theta_t; \Theta_{0:t-1}, Z_{\ell,0:t}) , \quad (6.7)$$

for some constant k , which implies

$$\pi'_t(\theta_t) \approx k' L_t(\theta_t; \Theta_0, Z_{L,[0,t]}) \ell_t(\theta_t; \Theta_{0:t-1}, Z_{\ell,0:t}) , \quad (6.8)$$

for some constant k' . So under the expected agreement between ℓ_t and L_t in this case, it is reasonable to assume that eq. (6.6) holds for most θ_t and θ'_t . The maximum magnitude of non-translational deformations between consecutive frames under which this conjectured behaviour holds could potentially be increased by using a subregion dissimilarity measure that has strong invariance to rotations and non-rigid deformations, such as the Earth Mover's Distance between pixel histograms (as opposed to our current calculation of the EMD between histograms of corresponding pixel differences), but we shall explore this in future work.

If we assume that eq. (6.6) holds for all θ_t and θ'_t , then it follows by definition of Ω'_t that

$$\begin{aligned} \forall \theta'_t, \theta_t \left(\theta'_t \in \Omega'_t, \theta_t \notin \Omega'_t \Rightarrow L_t(\theta_t; \Theta_0, Z_{L,[0,t]}) \geq L_t(\theta'_t; \Theta_0, Z_{L,[0,t]}) \right) \\ \Leftrightarrow \forall \theta_t \left(\theta_t \notin \Omega'_t \Rightarrow L_t(\theta_t; \Theta_0, Z_{L,[0,t]}) \geq \tau'_t \right) , \end{aligned} \quad (6.9)$$

for some threshold $\tau'_t > 0$. From this, it follows that there is a threshold $\tau_t > 0$ that satisfies

$$\Delta_{L,0,t}[\mathbf{V}(\Theta_0), \mathbf{V}(\Theta_t^{[i]})] > \tau_t \Rightarrow \Theta_t^{[i]} \in \Omega'_t , \quad (6.10)$$

where $\Delta_{L,0,t}$ is the image dissimilarity function that we defined in chapter 4. The falsehood of the antecedent here is not generally a sufficient condition for us to conclude the converse, that $\Theta_t^{[i]} \in \Omega - \Omega'_t$, and it would not be easy to determine the smallest value of τ_t for which this test is true for all $\Theta_t^{[i]}$, but this idea forms the basis for the loss test that we have developed.

Let

$$W_t^{[i]} \triangleq \Theta_t^{[i]} \in \Omega'_t \quad (6.11)$$

be a boolean random variable that takes value \top when the i^{th} particle is lost at time t , and

value \perp otherwise. Also, for notational brevity let

$$\Delta_{\mathbf{L},t}^{[i]} \triangleq \Delta_{L,0,t}[\mathbf{V}(\Theta_0), \mathbf{V}(\Theta_t^{[i]})] \quad (6.12)$$

be a random variable that is completely determined given $\mathbf{Z}_{[0,t]}$ and $\Theta_{[0,t]}^{[i]}$. We tackle the particle loss detection problem in a hypothesis testing framework, in which the aim is to use our observation of $\Delta_{\mathbf{L},t}^{[i]}$ to compare the likelihood of the null hypothesis

$$H_0 : \mathbf{W}_t^{[i]} = \top \quad (6.13)$$

to that of the alternative

$$H_1 : \mathbf{W}_t^{[i]} = \perp . \quad (6.14)$$

We test the null hypothesis using a variant of the classic likelihood ratio test. As described in [17], the likelihood ratio test is usually used to compare the likelihood of a null hypothesis H'_0 that a parameter x belongs to some set X' to the likelihood of an alternative hypothesis H'_1 that x belongs to a set X that contains X' . Given a set of data samples D , the test rejects H'_0 when

$$\frac{\sup_{y \in X'} P(D|x=y)}{\sup_{y \in X} P(D|x=y)} \leq c , \quad (6.15)$$

for some constant $c \in [0, 1]$ used to control the probabilities of type I and type II errors. In our case, H_0 and H_1 are disjoint hypotheses, so we allow c to be any value > 0 , and we reject H_0 if the test

$$\frac{B_{G,t}(\Delta_{\mathbf{L},t}^{[i]})}{F_{G,t}(\Delta_{\mathbf{L},t}^{[i]})} \leq c \quad (6.16)$$

is true, where $B_{G,t}$ and $F_{G,t}$ are functions that we refer to as the *background likelihood* and the *foreground likelihood* respectively. We define them as

$$\begin{aligned} B_{G,t}(\Delta_{\mathbf{L},t}^{[i]}) &\triangleq P(\Delta_{\mathbf{L},t}^{[i]} \mid \mathbf{W}_t^{[i]} = \top, \mathbf{Z}_{[0,t]}, \Theta_0) \\ F_{G,t}(\Delta_{\mathbf{L},t}^{[i]}) &\triangleq P(\Delta_{\mathbf{L},t}^{[i]} \mid \mathbf{W}_t^{[i]} = \perp, \mathbf{Z}_{[0,t]}, \Theta_0) . \end{aligned} \quad (6.17)$$

When the test is false, we conclude that there is insufficient evidence in favour of H_1 , and so

we assume H_0 .

In the next section, we will discuss our formulation of the background and foreground likelihood functions, and the methods by which we estimate their parameters, the parameters of the importance sampling distribution ℓ_t and the patch likelihood function L_t , and the optimal value of the test threshold c .

6.4 Parameterising the Particle Loss Test

6.4.1 DTM-Simulation-Based Sample Set Creation

All four of the likelihood functions we have discussed so far – the background likelihood $B_{G,t}$, foreground likelihood $F_{G,t}$, patch likelihood L_t and local likelihood ℓ'_t – are defined as functions of differences between image regions measured by the weighted Euclidean dissimilarity function \mathcal{E} that we defined in chapter 4. When we defined L_t in terms of the stretched exponential function φ , we implicitly assumed it to be both spatially and temporally stationary in the sense that for any times t and t' and patch states θ and θ' , the random variable

$$\Delta_t = \Delta_{L,0,t}[\mathbf{V}(\Theta_0), \mathbf{V}(\theta)] \quad (6.18)$$

(that is random due to its dependence on $\mathbf{Z}_{L,t}$) follows the same distribution as the random variable

$$\Delta_{t'} = \Delta_{L,0,t'}[\mathbf{V}(\Theta_0), \mathbf{V}(\theta')] \quad (6.19)$$

(that is random due to its dependence on $\mathbf{Z}_{L,t'}$). This follows from the fact that for all δ

$$\begin{aligned} P(\Delta_t = \delta \mid \mathbf{Z}_{L,0}, \Theta_0, \Theta_t = \theta) &= L_t(\Theta_t = \theta; \Theta_0, \mathbf{Z}_{L,0}, \Delta_t = \delta) \quad (\text{by definition}) \\ &= \varphi(\delta; \alpha_L, \gamma_L) \\ &= L_{t'}(\Theta_{t'} = \theta'; \Theta_0, \mathbf{Z}_{L,0}, \Delta_{t'} = \delta) \\ &= P(\Delta_{t'} = \delta \mid \mathbf{Z}_{L,0}, \Theta_0, \Theta_{t'} = \theta') \quad , \end{aligned} \quad (6.20)$$

where, with slight abuse of notation, we pass Δ_t and $\Delta_{t'}$ to L_t and $L_{t'}$ to signify the amounts by which the regions of images $Z_{L,t}$ and $Z_{L,t'}$ within $V(\theta)$ and $V(\theta')$ differ from the region of Z_0 within $V(\Theta_0)$. The spatiotemporal stationarity of ℓ'_t follows by a similar argument, since under our model, the hyperparameters $\alpha_{\ell'}$ and $\gamma_{\ell'}$ that define the distribution of the subregion difference random variable $\Delta_{\ell'}$ do not vary with time or with the states of the patches that are being compared. We extend these stationarity assumptions to the background and foreground likelihoods, and to emphasise this we will drop their t subscripts from now on.

We divide the process of parameterising B_G , F_G , L_t and ℓ'_t into the following two steps:

1. Construct sets D_B , D_F , D_L and $D_{\ell'}$ of image differences that follow the distribution that we want each likelihood function to represent.
2. Identify the hyperparameters of the likelihood functions that maximise the likelihoods of the D_* sample sets.

We shall address the construction of the D_* sets in this section, and deal with the maximum-likelihood estimation of the likelihood function hyperparameters in section 6.4.3.

For all t , let $\nu_t^*(\mathbf{p})$ be a deformation field that perfectly maps each pixel \mathbf{p} in image Z_0 to the corresponding pixel in Z_t , and define Θ_t^* such that $\mathcal{M}(\cdot; \Theta_0, \Theta_t^*)^*$ is the best affine approximation of the transformation that ν_t^* induces on each point in the initial patch $V(\Theta_0)$. By the assumed stationarity of L_t , we could then define D_L as some subset of

$$\{\Delta_{L,0,t}[V(\Theta_0), V(\Theta_t^*)] : t > 0\} . \quad (6.21)$$

As ν^* is unknown however, we create an approximation of this set using the DTM. To do this, we sample n_L deformation trajectories from the DTM, sample each one at $T_L + 1$ uniformly-spaced phases, and use the resulting deformations $\nu_{L,1:n_L,0:T_L}$ to warp the initial lightness-normalised image and occlusion mask. For each deformation trajectory $\mathbf{q}^{(k)}$, we would ideally then calculate

* \mathcal{M} is the affine coordinate transformation that we defined on page 53.

the T_L best-fitting affine transformation parameters $\Theta'_{1:T_L}^{(k)}$, and then add the set

$$\left\{ \Delta_{L,\nu_L,k,[0,t]}[\mathbf{V}(\Theta_0), \mathbf{V}(\Theta_t'^{(k)})] : 1 \leq t \leq T_L \right\} \quad (6.22)$$

to D_L , where $\Delta_{L,\nu_L,k,[0,t]}$ is the result of applying $\Delta_{L,0,t}$ to the time 0 and t warped images produced by the k^{th} ν_L (we will use similar notation later with other deformation field sequences sampled from the DTM). However, we found that this process tended to underestimate the standard deviation of D_L , leading to numerically unstable underestimates of the γ_L inverse decay rate parameter of L_t . The main reasons for this seem to be that the $\Theta'_{1:T_L}^{(k)}$ fit the deformations generated by the DTM too well, and that we have not modelled all of the non-deformational appearance changes that occur on the myocardium, such as changes due to noise and changes of colour/texture that occur as the surface stretches and contracts. Some of the colour/texture changes look like the result of surface features fading, giving way to the prevailing colour of their surroundings, or colours in one region flowing into another (figure 6.1 gives some examples of these changes). We compensate for the DTM's inability to produce these effects by generating m_L small, zero-mean Gaussian-distributed perturbations $\delta\Theta_t^{(k,1:m_L)}$, and setting

$$D_L = \left\{ \Delta_{L,\nu_L,k,[0,t]}[\mathbf{V}(\Theta_0), \mathbf{V}(\Theta_t'^{(k)} + \delta\Theta_t^{(k,j)})] : 1 \leq k \leq n_L, 1 \leq j \leq m_L, 1 \leq t \leq T_L \right\} \quad , \quad (6.23)$$

where $\Theta_t'^{(k)} + \delta\Theta_t^{(k,j)}$ represents the result of perturbing $\Theta_t'^{(k)}$ by $\delta\Theta_t^{(k,j)}$. We will make all of this more precise in the next section.

The variations in $\Delta_{L,t}^{[i]}$ that F_G describes are the result of $\Theta_t^{[i]}$'s variations over $\Omega - \Omega'_t$, so once again, we measure patch differences with respect to n_F simulated sequences of $T_F + 1$ deformations $\nu_{F,1:n_F,0:T_F}$. Due to the difficulty of determining Ω'_t 's boundary after choosing a probability p that defines it, we instead assume that for some unknown p , $\Omega - \Omega'_t$ is well approximated by a neighbourhood $\mathcal{N}_t^{(k)}$ of $\Theta_t'^{(k)}$. Clearly, increasing $\mathcal{N}_t^{(k)}$'s size will have the effect of decreasing the p that corresponds to the $\Omega - \Omega'_t$ that $\mathcal{N}_t^{(k)}$ best fits.

The ideal way to generate D_F 's samples would be the rejection sampling approach of running the particle filter repeatedly and adding $\Delta_{L,\nu_F,k,[0,t]}[\mathbf{V}(\Theta_0), \mathbf{V}(\Theta_t^{[i]})]$ to D_F whenever $\Theta_t^{[i]} \in \mathcal{N}_t^{(k)}$.

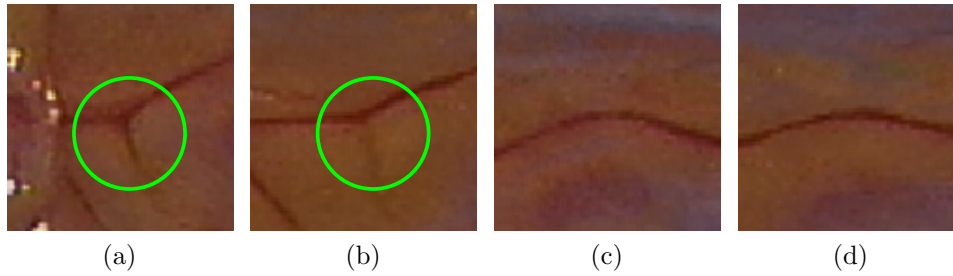


Figure 6.1: The encircled regions of (a) and (b) are an example of the kind of appearance change in which the colour of a feature (the near-vertical vessel section in (a)) fades away. (c) and (d) show some other changes of colour/texture that the DTM cannot generate. In some regions, colours seem to flow into each other, but in others new colours seem to appear from nowhere. Constructing D_L by perturbing $\Theta_t^{(k)}$ can account for the former of these occurrences to an extent, but it cannot account for the latter.

All four images were taken from the 230108.FO1 sequence, after lightness normalisation. (a) and (c) are from frame 11 of the sequence, and (b) and (d) are from frame 21.

But this would be impractical, especially considering the fact that the probability of the particles falling in each $\mathcal{N}_t^{(k)}$ would generally decrease as t increases. So instead, we use rejection sampling to draw m_F patch state samples from the region of a $\Theta_t^{(k)}$ -centred Gaussian distribution that lies within $\mathcal{N}_t^{(k)}$, adding the corresponding Δ_L values to D_F . Note that when the patch is very easy to distinguish from its surroundings, the neighbourhoods that correspond to reasonably small values of p should all be small, and so we can set $D_F = D_L$.

Similarly to F_G , B_G describes the variations in $\Delta_{L,t}^{[i]}$ that result from $\Theta_t^{[i]}$'s variations over Ω'_t . We have no prior reason to assume that any patch state in Ω'_t is more likely than another, so we define D_B by drawing m_B uniformly-distributed state samples from each of the T_B frames of n_B simulated image sequences, using rejection sampling to ensure that the samples lie outside $\mathcal{N}_t^{(k)}$ and within the range of transformations that the importance sampler can generate for a particle in state $\Theta_{t-1}^{(k)}$ at each time $t - 1$ (we defined this range in section 3.3.3).

The local likelihood functions ℓ'_t , that multiplicatively combine to define the importance sampler ℓ_t , represent the variations in the differences $\Delta_{\ell',t}$ between the image-boundary-aligned subregions $S(\mathbf{g}_{t-1})$ of $\mathbf{V}(\Theta_{t-1})$ and the corresponding image-boundary-aligned subregions $S(\mathbf{g})$ of $\mathbf{V}(\Theta_t)$, where each subregion centre \mathbf{g}_{t-1} is a grid vertex in the set G_{t-1} of vertices within $\mathbf{V}(\Theta_{t-1})$, calculated as described in section 3.3.1, and $\mathbf{g} = \mathcal{M}(\mathbf{g}_{t-1}; \Theta_{t-1}, \Theta_t)$. So again, we use

$n_{\ell'}$ sequences of $T_{\ell'}$ images $\mathbf{Z}'_{1:n,0:T}$ to generate samples for $D_{\ell'}$, constructing the set as follows:

$$D_{\ell'} = \left\{ \Delta_{\ell', \nu_{\ell', k, [t-1, t]}} \left[S(\mathbf{g}'_{t-1}), S \left(\mathcal{M} \left(\mathbf{g}'_{t-1}; \Theta'_{t-1}, \Theta'_t + \delta\Theta_t^{(k)} \right) \right) \right] : \right. \\ \left. 1 \leq k \leq n_{\ell'}, \mathbf{g}'_{t-1} \in G_{t-1}^{(k)}, 1 \leq t \leq T_{\ell'} \right\}, \quad (6.24)$$

where $G_{t-1}^{(k)}$ is the set of grid vertices within $\mathbf{V}(\Theta_{t-1}^{(k)})$, $\Delta_{\ell', \nu_{\ell', k, [t-1, t]}}$ is $\Delta_{\ell', t}$ applied to images warped by the simulated deformations $\nu_{\ell', k, [t-1, t]}$, and $\delta\Theta_t^{(k)}$ is a Gaussian perturbation that we use once again to account for colour changes.

6.4.2 Simulation Details

We begin the generation of a simulated image sequence by sampling a trajectory vector \mathbf{p} from a B-Spline-Based Gaussian DTM. To generate \mathbf{p} using the M largest modes of the DTM, we generate an M -vector $\boldsymbol{\epsilon}$ of independent standard Gaussian random variables using the Box-Muller transform ([15]), and set \mathbf{p} to

$$\mathbf{p} = \bar{\mathbf{p}} + \mathbf{E}_{:,1:M} \text{diag}(\lambda_1^{\frac{1}{2}}, \dots, \lambda_M^{\frac{1}{2}}) \boldsymbol{\epsilon}, \quad (6.25)$$

where $\bar{\mathbf{p}}$ is the model's mean trajectory vector, λ_i is its i^{th} largest eigenvalue, and $\mathbf{E}_{:,1:M}$ is the matrix of M corresponding eigenvectors.

Before we can use the deformation trajectory $\mathbf{q}(\cdot; \mathbf{p})$ defined by \mathbf{p} to deform the initial images, we must recalculate/resample the deformation sequence properties that were removed during construction of the model, namely the period T of the cardiac cycle (in frames), the phase \mathbf{o} at which to begin the sequence, and a 2D similarity transformation

$$f(\mathbf{x}; \rho, \phi) = \rho \mathbf{R}(\phi) \mathbf{x} + \mathbf{d}(\Theta_0) \quad (6.26)$$

that maps the sequence of deformed grids defined by \mathbf{q} from the canonical grid's coordinate system to that of the patch's initial state $\mathbf{V}(\Theta_0)$, where $\mathbf{R}(\phi)$ is an anticlockwise rotation by ϕ radians, $\mathbf{d}(\Theta_0)$ is the initial position of the patch, and ρ is a scaling factor. The rotation

parameter is arbitrary, so we uniformly sample it over $[0, 2\pi)$.

Given these properties, an image \mathbf{Y}_t can be generated from an initial image \mathbf{Y}_0 using standard graphics library texturing functions. In particular, letting

$$(\mathbf{h}'_t{}^{-1}, \mathbf{g}'_t) = \Upsilon^+ \left(\mathbf{q} \left(\mathbf{o} + \frac{t}{T+1}; \mathbf{p} \right) \right) \quad (6.27)$$

denote the time t inverse affine transformation $\mathbf{h}'_t{}^{-1}$ and non-affine deformed canonical grid \mathbf{g}'_t (as calculated by the mapping Υ^+ that we defined in section 5.4.3), we generate texture coordinates that map the vertices of the initial grid $f(\mathbf{h}'_0{}^{-1}(\mathbf{g}'_0))^*$ to the corresponding pixels of \mathbf{Y}_0 , and then we generate the time t image \mathbf{Y}_t by rendering the quadrilaterals of $f(\mathbf{h}'_t{}^{-1}(\mathbf{g}'_t))$, assigning the initial texture coordinates to the corresponding vertices of this new grid, so that the motion of the vertices warps the initial image. We set each occlusion mask $\mathcal{F}_{O,t}$ to 1 everywhere outside grid $f(\mathbf{h}'_t{}^{-1}(\mathbf{g}'_t))$, indicating that the pixels outside the grids are undefined. Figure 6.2 shows an example of this process rendered with OpenGL.

Our tests at the end of this chapter were all performed offline, so we defined T manually. But for general use, it could be determined by a number of methods. The simplest method would of course be to obtain the information from an ECG. But it may be preferable to measure it directly from the video in some instances, such as when an ECG is not available. A simple method that we have found to be reasonably effective in the past when dealing with videos that contain no camera motion is to calculate the autocorrelation of the pixels within a fixed region of the images, and take T to be the period between the peaks in the autocorrelation signal. A more sophisticated and robust method, that can handle camera motion, intrinsic camera parameter changes and affine respiratory motion, but requires a set of landmarks to have already been tracked, was proposed in [50]. It uses the epipolar constraint to identify time points at which the landmark sets are equal up to a projective transformation of their original unknown 3D positions, and hypothesises that these time points must occur at the same phase of each cycle.

*We use $f(\mathbf{h}'_t{}^{-1}(\mathbf{g}'_t))$ to indicate the application of f to every point in $\mathbf{h}'_t{}^{-1}(\mathbf{g}'_t)$.

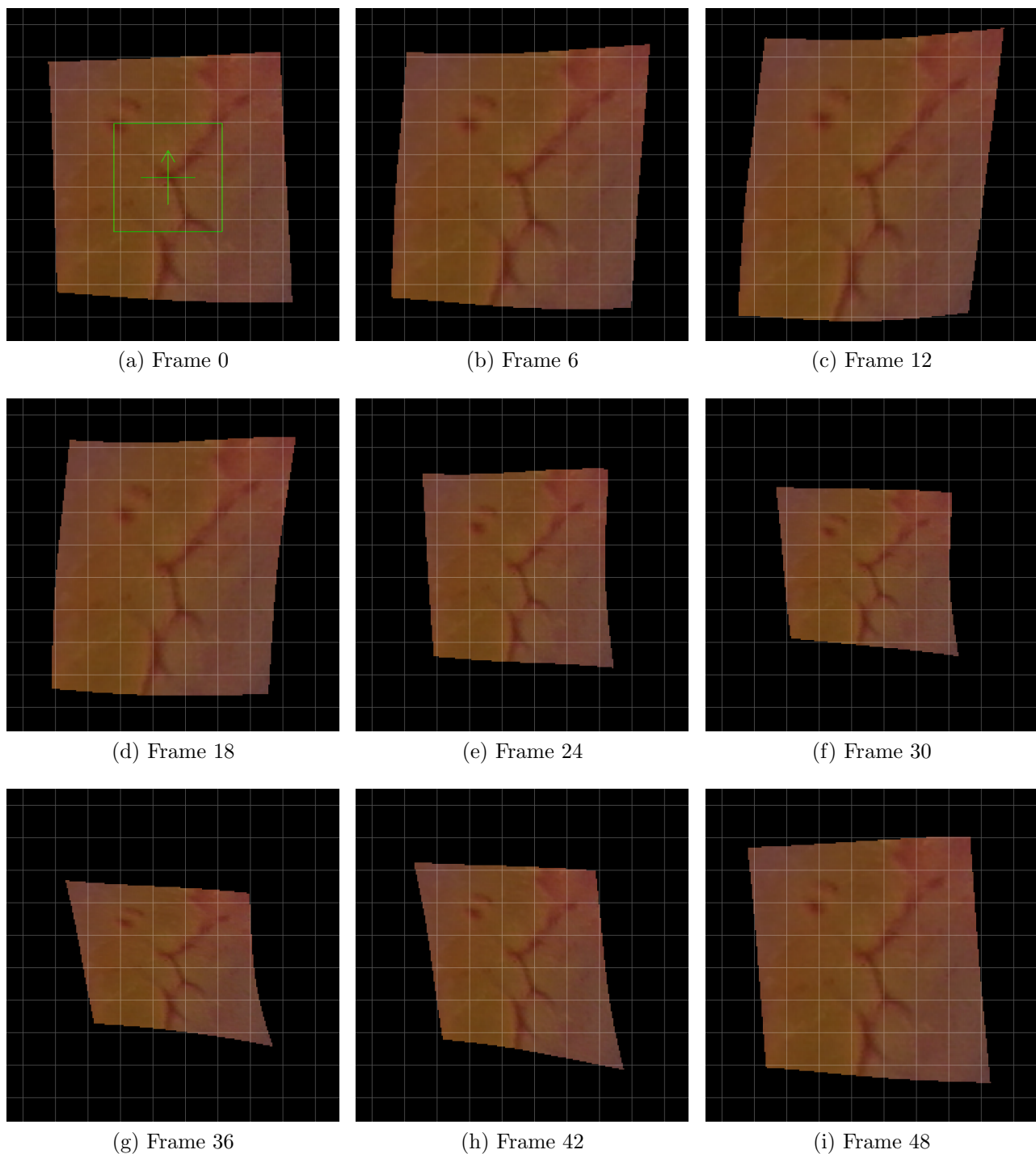


Figure 6.2: Some frames from a 50 frame deformation sequence sampled from the DTM that we constructed for the 180608.HS sequence and applied to the region of the first image surrounding the initial feature state shown in (a). For reference, we have rendered a regular grid with 24-pixel-wide squares over each image.

Calculating the Starting Phase

As the scaling components of the similarity transformations used in the construction of the deformation mode model defined the amounts by which the canonical grid \mathbf{g} should be scaled, we must choose a starting phase \mathbf{o} at which the shape of the grid defined by the deformation trajectory is as close as possible to \mathbf{g} 's shape, so as to ensure that the range over which we sample ρ is comparable to the scaling factor range used in the deformation mode model. The easiest way to identify such a starting phase is to search for one of the deformation trajectory's candidate phase offsets, which we defined in section 5.4.3 as the minimisers of $x(o') \triangleq \|\mathbf{q}(o'; \mathbf{p})\|^2$. We can efficiently estimate a candidate phase offset by Newton-Raphson minimisation, using the fact that the first two derivatives $x^{(1)}$ and $x^{(2)}$ of x satisfy

$$\begin{aligned} x^{(1)}(o') &= 2\mathbf{q}(o'; \mathbf{p})^T \mathbf{q}^{(1)}(o'; \mathbf{p}) \\ &= 2\mathbf{b}(o')^T [\mathbf{p}]^T [\mathbf{p}] \mathbf{b}^{(1)}(o') \\ x^{(2)}(o') &= 2\mathbf{b}^{(1)}(o')^T [\mathbf{p}]^T [\mathbf{p}] \mathbf{b}^{(1)}(o') + 2\mathbf{b}(o')^T [\mathbf{p}]^T [\mathbf{p}] \mathbf{b}^{(2)}(o') . \end{aligned} \quad (6.28)$$

[81] gives the following useful formula for the first derivative of the degree d B-spline blending function $b_{d,i}(o')$:

$$b_{d,i}^{(1)}(o') = \frac{d}{t_{i+d} - t_i} b_{d-1,i}(o') - \frac{d}{t_{i+d+1} - t_{i+1}} b_{d-1,i+1}(o') , \quad (6.29)$$

from which it follows that

$$b_{d,i}^{(2)}(o') = \frac{d}{t_{i+d} - t_i} b_{d-1,i}^{(1)}(o') - \frac{d}{t_{i+d+1} - t_{i+1}} b_{d-1,i+1}^{(1)}(o') , \quad (6.30)$$

where the t_i are the B-spline knots. Given an initial estimate o'_0 , the Newton-Raphson algorithm calculates an improvement o'_n of the $(n-1)^{\text{th}}$ estimate o'_{n-1} as

$$\begin{aligned} o'_n &= o'_{n-1} + \delta_{n-1} \\ \delta_{n-1} &= -\frac{x^{(1)}(o'_{n-1})}{x^{(2)}(o'_{n-1})} . \end{aligned} \quad (6.31)$$

In cases where x is locally well-approximated by a quadratic function around o'_{n-1} , the algorithm generally converges rapidly. But in other cases, the step δ_{n-1} may be too large. So at each iteration we set o'_n to an approximate local minimiser $o'_{n-1} + \delta'_{n-1}$ of x , where $\delta'_{n-1} \leq |\delta_{n-1}|$. We search for δ'_{n-1} using a simple dichotomous line search method (described, for example, in [8]), which is a form of binary search in which one half of the search space (which is initially set to δ'_{n-1} 's range) is discarded in each iteration if x 's value at a point in that half near the midpoint is greater than its value near the midpoint in the other half. More sophisticated and efficient line search methods could have been used (see [73] for a comprehensive review), but we achieved very fast results with the dichotomous line search, and it was simpler to implement than the alternatives.

We define multiple initial values for the search by selecting $d - 1$ values uniformly-distributed over each knot span $[t_i, t_{i+1})$. This is the minimum number that should be used over each span, since the degree d polynomial piece that each span corresponds to may have up to $d - 1$ turning points in each dimension. From the results we obtained, it did not seem to be necessary to sample each knot span more densely than this.

We use relative measures of convergence to determine when to terminate the both the line search and the Newton-Raphson minimisation. For the line search, suppose the search space has been reduced to an interval $[o'^{\downarrow}, o'^{\uparrow}]$ after the k^{th} binary subdivision. We calculate the maximum difference d_k between the values of x at o'^{\downarrow} , $\frac{o'^{\downarrow} + o'^{\uparrow}}{2}$ and o'^{\uparrow} , and we consider the search to have converged when d_k is less than the product of some threshold τ and the mean value of $d_{1:k-1}$. For the Newton-Raphson minimisation, we consider convergence to have occurred when $|x(o'_n) - x(o'_{n-1})|$ is less than the product of some threshold τ' and $|x(o'_n)|$. In both cases, upper bounds must be placed on the number of iterations that can be carried out, to prevent infinite loops. We found 30 iterations to be sufficient with $\tau = \tau' = 10^{-5}$, and in many cases convergence occurred in far fewer iterations.

Sampling the Scaling Factor

Given a candidate phase offset \mathbf{o} , let $a_{\mathbf{o}}$ be the area of grid $\mathbf{q}(\mathbf{o}; \mathbf{p})$ in pixels² (calculated by adding up the areas of all of the grid's quadrilaterals), and let $[a^{\downarrow}, a^{\uparrow}]$ be the range of areas (also in pixels²) that corresponds to the range areas of the similarity-transformed canonical grids that were used in the construction of the deformation mode model. When the DTM is being used simulate deformations of images from the video that it was constructed from (as in the tests at the end of this chapter), these two ranges of areas are equal. But in the more general case, in which a DTM is constructed from multiple patient videos before being applied to a new video, the former range of areas would have to be determined from some depth-invariant measure of the areas of regions within the training videos and the new video. One possible way to define such a measure would be to consider the size of myocardial features that can be recognised across patients. Better yet, if optical flow fields could be reliably calculated between consecutive frames, the ranges of displacements across regions of the flow fields (the range over a region may be a useful indicator of area since the range of displacements over a region can be no less than the range over its subregions) could be exploited, calculating the ranges relative to the sizes of the regions in pixels, to obtain measures of area that have a degree of invariance to the camera's distance from the myocardium.

The squared scaling factor ρ^2 should be uniformly sampled over a range defined such that the area of the initial grid $f(\mathbf{h}'_0^{-1}(\mathbf{g}'_0))$ always lies within $[a^{\downarrow}, a^{\uparrow}]$, which implies that ρ^2 's range should be a subset of $[\frac{a^{\downarrow}}{a_{\mathbf{o}}}, \frac{a^{\uparrow}}{a_{\mathbf{o}}}]$. In addition to this, we would also ideally like grid $f(\mathbf{h}'_t^{-1}(\mathbf{g}'_t))$ to contain as much of the area occupied by the patches used in the construction of the D_* set as possible, for each time step t . However, it would not be easy to identify the smallest value of ρ^2 for which each $f(\mathbf{h}'_t^{-1}(\mathbf{g}'_t))$ at least contains a pre-specified proportion of the area of each patch, due to the non-analytical relationship between the value of ρ and the state of the best-fitting affine transformation Θ'_t . So instead, we use the simple heuristic of ensuring that ρ is large enough for the initial grid to contain as many time $t = 1$ patches used in the construction of D_* as possible, subject to the first constraint on ρ^2 above being satisfied. This involves identifying the shortest distance $d_{\mathbf{o}}^{\downarrow}$ from the origin to a vertex of $\mathbf{q}(\mathbf{o}; \mathbf{p})$, and an upper bound m^{\uparrow} on the

distances between the initial feature position $\mathbf{d}(\Theta_0)$ and the possible positions of the vertices of the time $t = 1$ patches, so that we may define ρ^2 's range as

$$\left[\min \left(\frac{a^\uparrow}{a_o}, \max \left(\frac{a^\downarrow}{a_o}, \left(\frac{m^\uparrow}{d_o^\downarrow} \right)^2 \right) \right), \frac{a^\uparrow}{a_o} \right] . \quad (6.32)$$

For D_F ($= D_L$), we specify each neighbourhood $\mathcal{N}_t^{(k)}$ as a set of bounds on the rotational, scaling, skewing and translational components that we will ultimately combine as in eq. (3.1) on page 42 to define the perturbed patch states. So given values for these bounds, we scale the initial patch $\mathbf{V}(\Theta_0)$ by the maximum amount permitted by $\mathcal{N}_1^{(k)}$ and try all four combinations of positively/negatively skewing it by the maximum amount in the x/y directions, and then we define m^\uparrow as the sum of the maximum distance from $\mathbf{d}(\Theta_0)$ to one of $\mathbf{V}(\Theta_0)$'s transformed vertices, and the maximum distance by which $\mathcal{N}_1^{(k)}$ allows the patch to be displaced. We define m^\uparrow similarly for $D_{\ell'}$, using a similar definition of the range of Gaussian perturbations, but we add the radius $\frac{2r_s+1}{\sqrt{2}}$ of the square subregions used in the definition of ℓ'_t as an extra term. Our definition of m^\uparrow for D_B is also similar to our definition of it for D_F , except that this time the bounds on the magnitudes of the affine motion components are given by the d^\uparrow , ϕ^\uparrow , κ_x^\uparrow , κ_y^\uparrow and s^\uparrow parameters that we used to define the domains of the importance sampling distributions in section 3.3.3.

Final Steps

After calculating/sampling the parameters of the similarity transformation f , the final steps are to calculate the affine transformations Θ'_t that best fit the deformation grids $f(\mathbf{h}'^{-1}(\mathbf{g}'_t))$, and to randomly perturb them. We estimate each Θ'_t using a 10×10 grid of points P regularly distributed over the initial patch $\mathbf{V}(\Theta_0)$. We deform P with $f(\mathbf{h}'^{-1}(\mathbf{g}'_t))$ (using bilinear interpolation, as described in section 5.2) to form a point set P' , and calculate the affine transformation \hat{f} that best maps P to P' using the Moore-Penrose pseudoinverse, as described in section 5.3.2. Finally, we define Θ'_t in terms of the translational component of \hat{f} and the *RSK* decomposition of its non-translational component.

To calculate a random perturbation $\Theta' + \delta\Theta$, we first of all draw samples $\delta\mathbf{d}$, $\delta\phi$, $(\delta\kappa_x, \delta\kappa_y)$ and $(\ln(\delta s_x), \ln(\delta s_y))$ from zero-mean uniform or Gaussian distributions, as appropriate (when using Gaussian distributions, we define the standard deviations to be a third of the sampling range), and we also sample a uniform binary random variable $\mathbf{b} \in \{0, 1\}$ that we use to decide whether to skew in the x or y direction. We then update Θ' by adding $\delta\mathbf{d}$, $\delta\phi$ and $(\mathbf{b}\delta\kappa_x, (1 - \mathbf{b})\delta\kappa_y)$ to its displacement, rotation and skewing components, and replacing its scaling components with $(\delta s_x \Theta'_{[s_x]}, \delta s_y \Theta'_{[s_y]})$.

6.4.3 Stretched Exponential Likelihood Parameterisation

The regions surrounding the initial patches, over which we calculate D_B , vary greatly, leaving us with no prior expectations about the form of the distribution of D_B 's values. So we construct the background likelihoods empirically, using a Gaussian kernel over D_B to estimate B_G :

$$B_G(\delta) = \sum_{\delta' \in D_B} N(\delta - \delta'; 0, \sigma) , \quad (6.33)$$

for some standard deviation σ . For efficiency, we use a discretisation of B_G that we calculate by convolving the Gaussian kernel with a histogram defined over D_B . We manually select a value for σ that removes local density fluctuations from the histogram while preserving its overall shape. Although this gave adequate results, it may be possible to derive better estimates of B_G 's values in low density areas using a variable bandwidth density estimator, such as one of the estimators described in [109]. We have not taken this more complex approach however, as we feel that the most pertinent way to improve the accuracy of our hypothesis tests would be to relax some of the assumptions/restrictions we have made in defining B_G and F_G , e.g. by using images other than the initial image as references.

While we make no prior assumptions about the form of the distribution over D_B , we do have prior expectations about the forms of the distributions over D_F/D_L and $D_{\ell'}$, so it is preferable to fit a parametric model that enforces these expectations to the distributions over these sets. Such models have added benefits of typically only having a small number of parameters to

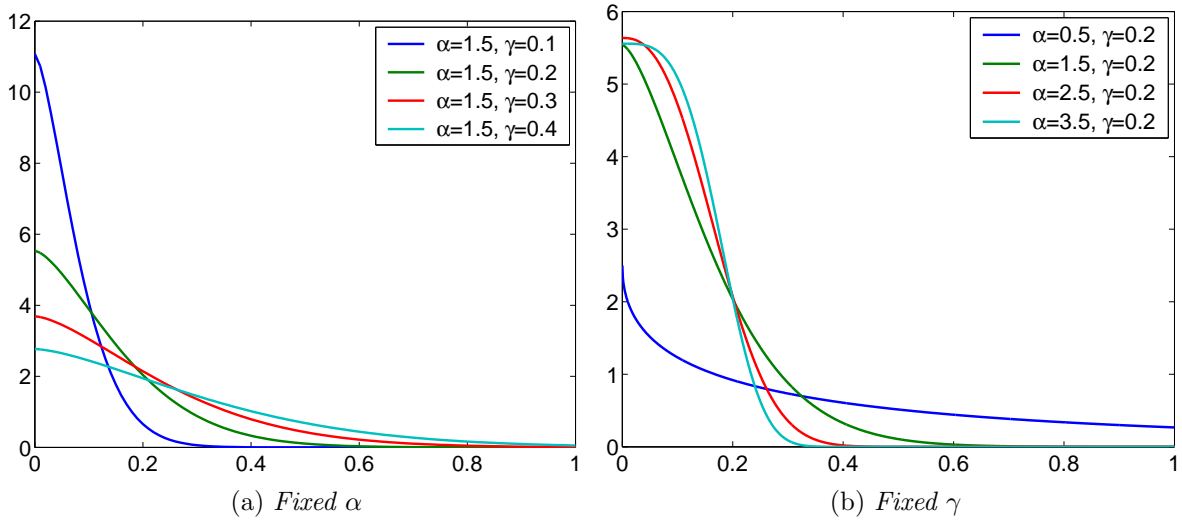


Figure 6.3: Stretched exponential distributions for various values of α and γ .

estimate, and alleviating the difficulties we would otherwise face in trying to estimate the probabilities of rare events.

The main prior assumption we wish to impose upon the distributions over D_F/D_L and $D_{\ell'}$ is that the likelihood of an image patch difference Δ is a continuous, strictly-decreasing function of Δ . As mentioned in section 3.3.1, the distribution with these properties that we choose to use for the importance sampler ℓ'_t and patch likelihood function L_t is the two-parameter stretched exponential distribution

$$\varphi(\Delta; \alpha, \gamma) \triangleq \exp \left\{ - \left(\frac{\Delta}{\gamma} \right)^\alpha \right\} \quad , \quad \alpha > 0, \gamma > 0 \quad , \quad (6.34)$$

which includes the exponential distribution and the single-tailed Gaussian distribution as special cases (when the stretching exponent α is set to 1 and 2, respectively. See figure 6.3 for examples of the distribution under various parameterisations). We assume that the foreground likelihood F_G is of the same form, since we define $D_F = D_L$, and so we only have to estimate two sets of hyperparameters. Whilst we were content to use the distribution in unnormalised form earlier, the normalising constant is vital to our present discussion, as our aim is to estimate the parameters by the maximum-likelihood method.

The stretched exponential distribution is a single-tailed special case of the generalised normal

distribution described in [71]. From the results published in that article, it follows that the normalised stretched exponential PDF $\hat{\varphi}(\Delta; \alpha, \gamma)$ is

$$\hat{\varphi}(\Delta; \alpha, \gamma) \triangleq \frac{\alpha \varphi(\Delta; \alpha, \gamma)}{\gamma \Gamma\left(\frac{1}{\alpha}\right)} , \quad (6.35)$$

and the CDF $\hat{\Phi}(\Delta; \alpha, \gamma)$ is

$$\hat{\Phi}(\Delta; \alpha, \gamma) \triangleq \frac{\Gamma\left(\frac{1}{\alpha}\right) - \Gamma\left(\frac{1}{\alpha}, \left(\frac{\Delta}{\gamma}\right)^\alpha\right)}{\Gamma\left(\frac{1}{\alpha}\right)} , \quad (6.36)$$

where $\Gamma(s, x)$ is the upper incomplete Gamma function:

$$\Gamma(s, x) \triangleq \int_x^\infty t^{s-1} e^{-t} dt , \quad (6.37)$$

and

$$\Gamma(s) = \Gamma(s, 0) \quad (6.38)$$

is the complete Gamma function.

Before calculating maximum likelihood estimates of the stretched exponential parameters, it is prudent to inspect the empirical distributions of the image patch difference sets D_* to see how close they are to stretched exponential distributions. Figure 6.4 shows typical examples of the empirical distributions. The most striking thing about these distributions is the fact that they do not peak at zero. This would almost certainly always happen regardless of whether the histograms were calculated under appearance changes caused by simulated deformations, or appearance changes caused by the true deformations, since the fact that the myocardium is in perpetual motion makes it very unlikely for any patch to have the exact same appearance (or extremely similar appearances) in many frames.

It is of course highly unlikely that any stretched exponential random variable could generate distributions like these that peak so far from zero, but an alternative model of F_G , L_t and ℓ'_t that gives greater likelihood to large patch differences than to small differences would be objectionable from a practical point of view, since it would bring about the absurd phenomenon

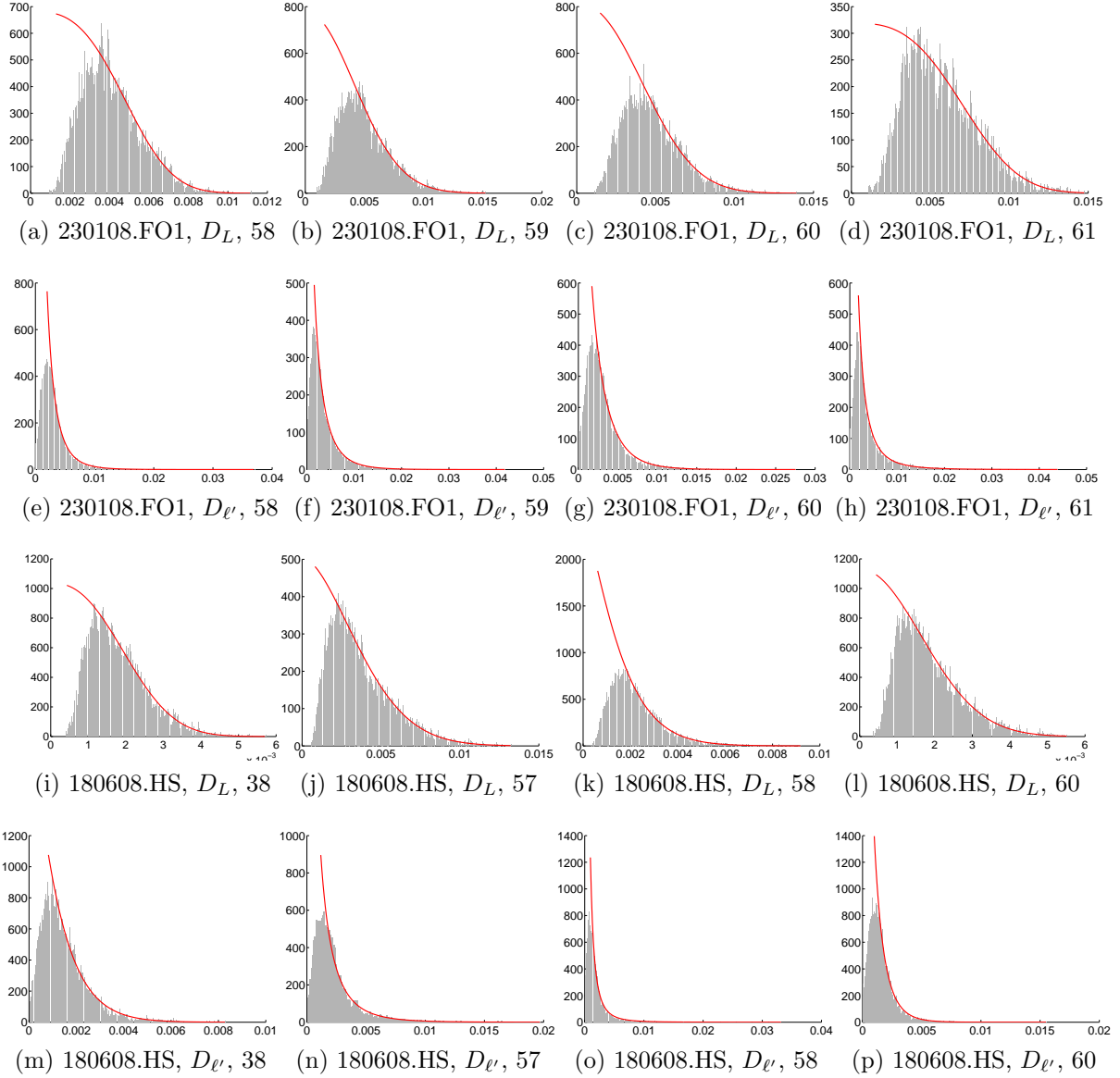


Figure 6.4: These figures show histograms for the D_L and $D_{\ell'}$ sample sets generated for some patches from the 230108.FO1 sequence and the 180608.HS sequence. The rightmost number underneath each figure is an identifier for the patches – figure 6.8 shows what these patches looked like. The stretched exponential distributions that maximise the conditional likelihood of each sample set are shown as (truncated) solid red lines.

of the particle filter shying away from rarely occurring states in which the patch looks very similar to its initial state! So for each sample set D_* , we calculate the stretched exponential hyperparameters α_* and γ_* that maximise the conditional likelihood of the elements of D_* that lie in the upper tail of its histogram, under the assumption that the rate of decay of the upper tail is a reasonable indicator of how fast the stretched exponential should tend to zero. Noting that the elements of D_* are always ≤ 1 , by definition of \mathcal{E} , we want to find

$$(\alpha_*, \gamma_*) = \arg \max_{\alpha > 0, \gamma > 0} \prod_{\delta \in D_*, \delta \geq \delta^\downarrow} L(\delta; \alpha, \gamma) , \quad (6.39)$$

for some lower bound δ^\downarrow on the difference samples, where

$$\begin{aligned} L(\delta; \alpha, \gamma) &= P(\Delta = \delta \mid \Delta \in [\delta^\downarrow, 1], \alpha, \gamma) \\ &= \frac{\hat{\varphi}(\delta; \alpha, \gamma)}{\hat{\Phi}(1; \alpha, \gamma) - \hat{\Phi}(\delta^\downarrow; \alpha, \gamma)} \\ &= \frac{\alpha \exp \left\{ - \left(\frac{\delta}{\gamma} \right)^\alpha \right\}}{\gamma \left(\Gamma \left(\frac{1}{\alpha}, \left(\frac{\delta^\downarrow}{\gamma} \right)^\alpha \right) - \Gamma \left(\frac{1}{\alpha}, \left(\frac{1}{\gamma} \right)^\alpha \right) \right)} . \end{aligned} \quad (6.40)$$

To simplify matters and make the problem amenable to being solved by standard minimisation algorithms, we take the usual approach of minimising the negative log-likelihood

$$(\alpha_*, \gamma_*) = \arg \min_{\alpha > 0, \gamma > 0} f(\alpha, \gamma) \quad (6.41)$$

where

$$\begin{aligned} f(\alpha, \gamma) &\triangleq - \ln \prod_{\delta \in D_*, \delta \geq \delta^\downarrow} L(\delta; \alpha, \gamma) \\ &= - \sum_{\delta \in D_*, \delta \geq \delta^\downarrow} \left\{ \ln \alpha - \left(\frac{\delta}{\gamma} \right)^\alpha - \ln \gamma - \ln h(\alpha, \gamma) \right\} \\ h(\alpha, \gamma) &\triangleq \Gamma \left(\frac{1}{\alpha}, \left(\frac{\delta^\downarrow}{\gamma} \right)^\alpha \right) - \Gamma \left(\frac{1}{\alpha}, \left(\frac{1}{\gamma} \right)^\alpha \right) \end{aligned} \quad (6.42)$$

(care must be taken when evaluating $\ln h$, as h may be rounded down to 0 when γ is very small). We turn this into an unconstrained optimisation problem by defining

$$\begin{aligned}\alpha &= e^a \\ \gamma &= e^g \ ,\end{aligned}\tag{6.43}$$

and minimising $f(e^a, e^g)$ with respect to a and g . We found MATLAB's implementation of the trust-region-based method described in [19, 20] to be an effective solution to the problem. This method works by iteratively minimising quadratic approximations of the objective function f over a neighbourhood of the current solution. The quadratic approximations are calculated from the gradients of f and from approximations of the Hessian, so we need to define the first derivatives of f with respect to a and g .

By repeated application of the chain rule, the derivatives can be expressed as follows:

$$\begin{aligned}\frac{\partial f(\alpha, \gamma)}{\partial a} &= e^a \frac{\partial f(\alpha, \gamma)}{\partial \alpha} \\ \frac{\partial f(\alpha, \gamma)}{\partial g} &= e^g \frac{\partial f(\alpha, \gamma)}{\partial \gamma} \\ \frac{\partial f(\alpha, \gamma)}{\partial \alpha} &= - \sum_{\delta \in D_*, \delta \geq \delta^\downarrow} \left\{ \frac{1}{\alpha} - \left(\frac{\delta}{\gamma} \right)^\alpha \ln \frac{\delta}{\gamma} - \frac{1}{h(\alpha, \gamma)} \frac{\partial h(\alpha, \gamma)}{\partial \alpha} \right\} \\ \frac{\partial f(\alpha, \gamma)}{\partial \gamma} &= - \sum_{\delta \in D_*, \delta \geq \delta^\downarrow} \left\{ \frac{\alpha}{\gamma} \left(\frac{\delta}{\gamma} \right)^\alpha - \frac{1}{\gamma} - \frac{1}{h(\alpha, \gamma)} \frac{\partial h(\alpha, \gamma)}{\partial \gamma} \right\} \\ \frac{\partial h(\alpha, \gamma)}{\partial \alpha} &= \frac{1}{\alpha^2} \left(\frac{\partial \Gamma \left(x, \left(\frac{1}{\gamma} \right)^\alpha \right)}{\partial x} \Big|_{x=\frac{1}{\alpha}} - \frac{\partial \Gamma \left(x, \left(\frac{\delta^\downarrow}{\gamma} \right)^\alpha \right)}{\partial x} \Big|_{x=\frac{1}{\alpha}} \right) \\ &\quad + \left(\frac{\delta^\downarrow}{\gamma} \right)^\alpha \ln \frac{\delta^\downarrow}{\gamma} \frac{\partial \Gamma \left(\frac{1}{\alpha}, y \right)}{\partial y} \Big|_{y=\left(\frac{\delta^\downarrow}{\gamma} \right)^\alpha} - \left(\frac{1}{\gamma} \right)^\alpha \ln \frac{1}{\gamma} \frac{\partial \Gamma \left(\frac{1}{\alpha}, y \right)}{\partial y} \Big|_{y=\left(\frac{1}{\gamma} \right)^\alpha} \\ \frac{\partial h(\alpha, \gamma)}{\partial \gamma} &= \frac{\alpha}{\gamma} \left(\frac{1}{\gamma} \right)^\alpha \frac{\partial \Gamma \left(\frac{1}{\alpha}, y \right)}{\partial y} \Big|_{y=\left(\frac{1}{\gamma} \right)^\alpha} - \frac{\alpha}{\gamma} \left(\frac{\delta^\downarrow}{\gamma} \right)^\alpha \frac{\partial \Gamma \left(\frac{1}{\alpha}, y \right)}{\partial y} \Big|_{y=\left(\frac{\delta^\downarrow}{\gamma} \right)^\alpha} \\ \frac{\partial \Gamma \left(\frac{1}{\alpha}, y \right)}{\partial y} &= -y^{\frac{1}{\alpha}-1} e^{-y} \ .\end{aligned}\tag{6.44}$$

Geddes et al. give a closed-form solution for $\frac{\partial \Gamma(x, y)}{\partial x}$ in [35] in terms of a very general special function known as the Meijer G function, which, under various parameterisations, reduces to

many more well-known special functions. Unfortunately, this function does not currently seem to be widely supported by numerical libraries, so we estimate the derivative by the central difference method instead:

$$\frac{\partial \Gamma(x, y)}{\partial x} \approx \frac{\Gamma(x + \delta x, y) - \Gamma(x - \delta x, y)}{2\delta x} , \quad (6.45)$$

for some small δx . We use a simple expression for δx given in [82] that approximately minimises the errors that result from truncation of the higher-order Taylor terms in the central difference approximation and the errors due to floating-point roundoff:

$$\delta x = \epsilon^{\frac{1}{3}} x , \quad (6.46)$$

where ϵ is the machine epsilon value – the smallest positive number for which $1 + \epsilon \neq 1$ in floating-point arithmetic.

To determine the lower bound δ^\downarrow , we convolve D_* 's histogram with a Gaussian kernel, and then set δ^\downarrow to the centre of the bin that has the greatest mass. We then initialise the minimiser by calculating f at values of α and γ distributed over a uniform grid, and setting the initial state to the grid point at which f is minimised.

6.4.4 Selecting the Test Threshold

The final parameter that we must define before we can carry out the loss hypothesis test is the comparison threshold c . Under the assumption that our parameterisations of B_G and F_G give reasonable approximations of $P(\Delta_{L,t}^{[i]} \mid \mathbf{W}_t^{[i]} = \top, Z_{[0,t]}, \Theta_0)$ and $P(\Delta_{L,t}^{[i]} \mid \mathbf{W}_t^{[i]} = \perp, Z_{[0,t]}, \Theta_0)$ respectively, we choose a value of c that minimises the probability of type I and type II test errors.

For any given choice of c , the probability of a type I error occurring, in which the test incorrectly

concludes that a lost particle is not lost, is given by integrating B_G over the set

$$E_I(c) = \{\delta \geq 0 : B_G(\delta) \leq cF_G(\delta)\} , \quad (6.47)$$

and the probability of a type II error, in which the test incorrectly concludes that a particle that is not lost is lost, is given by integrating F_G over

$$E_{II}(c) = \{\delta \geq 0 : B_G(\delta) > cF_G(\delta)\} . \quad (6.48)$$

We define c as the minimiser of

$$f(c') \triangleq \int_{E_I(c')} B_G(\delta) d\delta + \int_{E_{II}(c')} F_G(\delta) d\delta , \quad (6.49)$$

which is proportional to the probability of either a type I error occurring or a type II error occurring, under the assumption that the prior probability of a particle being lost is equal to the prior probability of it not being lost, i.e. assuming

$$P(\mathbf{W}_t^{[i]} = \top \mid \mathbf{Z}_{[0,t]}, \Theta_0) = P(\mathbf{W}_t^{[i]} = \perp \mid \mathbf{Z}_{[0,t]}, \Theta_0) . \quad (6.50)$$

This is probably not an accurate prior assumption, but it is the most noncommittal assumption we can make without further analysis.

Suppose we have defined B_G over N bins with boundaries $[b^\downarrow, b^\downarrow + w), \dots, [b^\downarrow + (N-2)w, b^\downarrow + (N-1)w), [b^\downarrow + (N-1)w, b^\uparrow = b^\downarrow + Nw]$, for some bin width w . Noting that $B_G(\delta) = 0 < cF_G(\delta)$ for all $\delta \notin [b^\downarrow, b^\uparrow]$ and $c > 0$, it follows that we only need to integrate the first term of f over $\delta \in [b^\downarrow, b^\uparrow]$. So we carry out the integration by adding up the integrals over each bin of B_G .

As we model F_G as a strictly-decreasing function and B_G as a histogram, which is a piecewise-uniform function, it is simple to calculate the integral over each bin. For any bin b , there are three possible cases to consider, as illustrated in figure 6.5. The cases are:

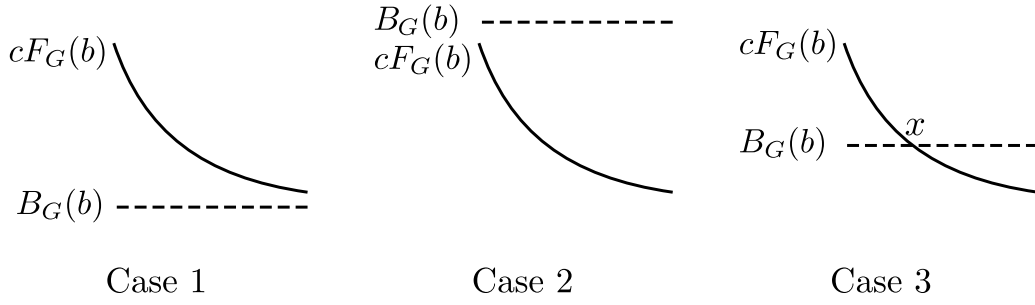


Figure 6.5: The three possible states of $B_G(b)$ relative to $cF_G(b)$, for some bin b and some value of c .

Case 1:

$$\forall \delta \in b \quad B_G(\delta) \leq cF_G(\delta) , \quad (6.51)$$

Case 2:

$$\forall \delta \in b \quad B_G(\delta) > cF_G(\delta) , \quad (6.52)$$

Case 3:

$$\exists x \in b \quad \forall \delta \in b \quad (\delta \leq x \rightarrow B_G(\delta) \leq cF_G(\delta) \wedge \delta > x \rightarrow B_G(\delta) > cF_G(\delta)) . \quad (6.53)$$

Let b_0 and b_1 be the lower and upper bounds of b . To determine whether or not the first case holds, it suffices to check whether or not $B_G(b_1) \leq F_G(b_1)$. In this case, $b \subseteq E_I(c)$, and so b 's contribution to f is

$$wB_G(b_0) . \quad (6.54)$$

Similarly, we can determine whether or not the second case holds by checking whether or not $B_G(b_0) > F_G(b_0)$. In this case $b \subseteq E_{II}(c)$, and b 's contribution to f is

$$\hat{\Phi}(b_1; \alpha_L, \gamma_L) - \hat{\Phi}(b_0; \alpha_L, \gamma_L) , \quad (6.55)$$

where $\hat{\Phi}$ is the stretched exponential CDF, as in the previous section, and α_L and γ_L are the parameters of the foreground/patch likelihood F_G/L_t . When neither the first case nor the second hold, we can determine the value of x under which the third case holds by using the

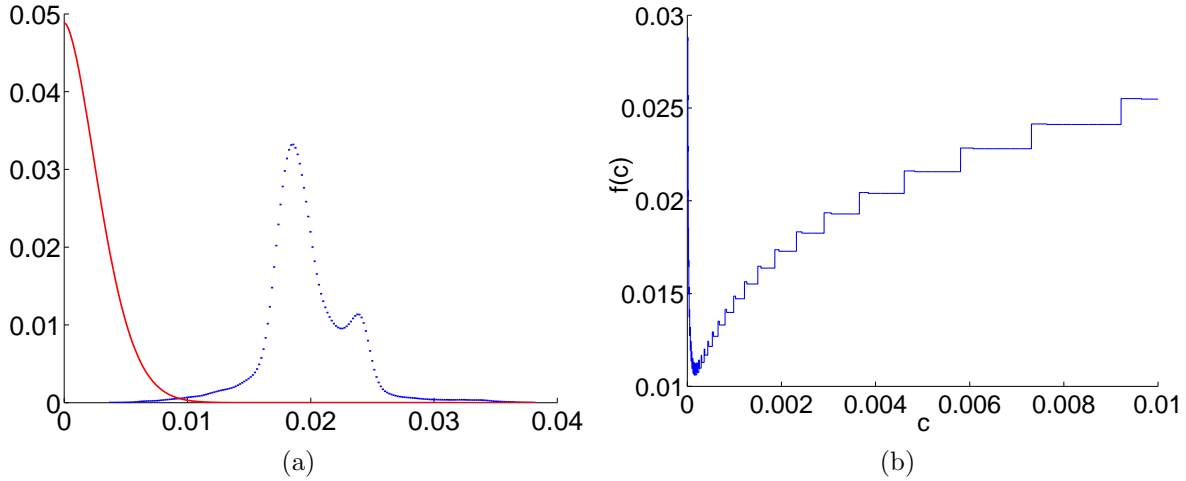


Figure 6.6: (a) The red curve shows $cF_G(\delta)$ as a function of δ for feature 35 of the 180608.HS sequence, using optimal values for c , α_L and γ_L , calculated as described in this section and the previous section. The discontinuous blue line segments show $B_G(\delta)$ for the same feature. (b) A graph of $f(c)$ for the same feature. f has numerous plateaus, which arise due to the fact that cF_G does not always intersect B_G .

inverse stretched exponential $\hat{\varphi}^{-1}$:

$$x = \hat{\varphi}^{-1} \left(\frac{B_G(b_0)}{c}; \alpha_L, \gamma_L \right)$$

$$\hat{\varphi}^{-1}(y; \alpha, \gamma) \triangleq \gamma \left(-\ln \left(\frac{y \gamma \Gamma(\frac{1}{\alpha})}{\alpha} \right) \right)^{\frac{1}{\alpha}} . \quad (6.56)$$

The section of b below x is in $E_I(c)$, and the upper section is in $E_{II}(c)$, so b 's contribution to f is given by

$$(x - b_0)B_G(b_0) + \hat{\Phi}(b_1; \alpha_L, \gamma_L) - \hat{\Phi}(x; \alpha_L, \gamma_L) . \quad (6.57)$$

f is a continuous function as, informally, small changes in c always induce small changes in $E_I(c)$ and $E_{II}(c)$, and the integrals of F_G and B_G vary continuously with the integration limits. However f is not differentiable everywhere, and it will often have many plateaus separated by extremely sharp transitions (e.g. see figure 6.6), and it may even have local minima, so gradient-based optimisation methods are not appropriate. Instead, we approximately minimise f using a variant of the dichotomous line search method that we described in section 6.4.2.

To begin with, we must calculate the bounds of a range $[c^\downarrow, c^\uparrow]$ over which to search for the

minimum. Despite the possibility of local fluctuations in f 's value, its overall form seems to be the same for most features – it initially undergoes a rapid decrease in value over very small values of c , and then its value increases gradually. We can easily identify a value of c^\downarrow within the range over which f rapidly decreases by simply selecting a value such that case 2 above holds for all bins. To do this, we calculate

$$c' = \min_{i, B_G(b^\downarrow + iw) > 0} \frac{B_G(b^\downarrow + iw)}{\hat{\varphi}(b^\downarrow + iw; \alpha_L, \gamma_L)} \quad , \quad (6.58)$$

and then set c^\downarrow to some value in $(0, c')$. The precise value is unimportant, and we arbitrarily choose a power of 2 that is $< c'$. We then need to identify a value of c^\uparrow that is large enough for the search range to contain the minimum. We set the initial candidate value c'' of c^\uparrow to c^\downarrow , and then perform an exponentially growing iterative search, each iteration of which involves evaluating $f(c'')$, keeping track of the candidate value c''^\downarrow at which f was minimised, and then replacing c'' with $2c''$. We terminate the search when $f(c'') \geq k f(c''^\downarrow)$ for some $k > 1$ (we found $k = 2$ to be sufficient), and set $c^\uparrow = c''$.

We then search $[c^\downarrow, c^\uparrow]$ for the minimising value of c much as before, with the exception that we cannot base our decision of which half of the search space to discard in each iteration on the values of f near the midpoint, due to the plateaus. So instead, we calculate f at m uniformly distributed values of c in each half, and discard the half that does not contain the minimum value of f over these sample points. We found $m = 4$ to be a sufficient number of sample points. We terminate the search either: when the maximum difference between f 's values at any pair of the sample points is no greater than the product of its maximum value at a sample point and a threshold τ ; or when the number of iterations exceeds some upper bound. We used $\tau = 1E^{-3}$ and allowed up to 100 iterations.

6.5 Some Methods for Restoring Lost Particles

Now that we have described the methods by which we estimate all of the parameters of our particle loss test, the final issue to be dealt with is the restoration of particles that our test

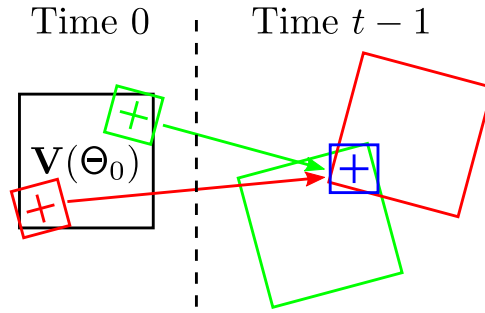


Figure 6.7: The green and red particles are to compare the blue time $t - 1$ subregion to time t subregions using image information from time 0. As the time $t - 1$ states of the particles are very different, the subregions of the first frame from which they will retrieve information (indicated by the small green and red squares) are also very different.

considers to be lost. The simplest method that we have tried is a resampling approach, in which the lost particles $\tilde{\Theta}_t^{[1:m]}$ are resampled from the particles $\Theta_t^{[1:n]}$ that are not marked as lost, in such a way that the resulting set of $m + n$ particles has the same weighted distribution as $\Theta_t^{[1:n]}$. To do this, we first of all associate each lost particle $\tilde{\Theta}_t^{[i]}$ with a particle $\Theta_t^{[f(i)]}$, where $f(i) \in \{1, \dots, n\}$ is a random variable that takes a value j with probability proportional to the weight $w_t^{[j]}$ of the j^{th} non-lost particle. After sampling all m values of f , we replace each $\tilde{\Theta}_t^{[i]}$ with $\Theta_t^{[f(i)]}$ and count the frequency $\lambda(j)$ with which f took value j , for each j . We then set the weights of particle $\Theta_t^{[j]}$ and the $\lambda(j)$ lost particles that were replaced with it to $\frac{w_t^{[j]}}{\lambda(j)+1}$.

While resampling has the expected effect of moving lost particles to states of high posterior probability, the resulting set of particles cannot be any more diverse than the original set of non-lost particles. Furthermore, resampling cannot be carried out at all when all particles are lost. An alternative particle restoration technique that does not suffer from these problems is to use information from earlier frames to define the importance sampling distribution ℓ_t . If information is used from a frame in which the state of the patch is known to a high degree of certainty, the cause of particle drift can potentially be avoided.

The frame in which the patch's state is known to the greatest degree of certainty will of course always be the first frame Z_0 , in which we define its initial state. As discussed earlier, we normally define ℓ_t by combining information about the differences $\Delta_{\ell',t}[S(\mathbf{g}), S(\mathbf{g}) + \mathbf{d}]$ between subregions $S(\mathbf{g})$ of image Z_{t-1} and $S(\mathbf{g}) + \mathbf{d}$ of image Z_t . So to make use of the first frame in the definition of ℓ_t for particle $\Theta_t^{[i]}$, we can use the projection of the image data from subregion

$\mathcal{M}(S(\mathbf{g}); \Theta_{t-1}^{[i]}, \Theta_0^{[i]})$ of image Z_0 into subregion $S(\mathbf{g})$ to calculate $\Delta_{\ell', t}$, instead of the information from Z_t .

Unfortunately, this restoration method has two significant limitations. Firstly, as shown in figure 6.7, the use of the first frame makes it possible for two different particles to end up projecting different parts of Z_0 into $S(\mathbf{g})$, which prevents us from caching the discrete, bounded-displacement local likelihood energy functions $\psi_t^{[i]}$ for later reuse. This causes a large performance penalty, particularly when all of a patch's particles have been labelled as lost. E.g. under the test data and parameters that we used for the tests that we will describe in the next section, using this method to restore 50 lost particles for each patch meant spending a few minutes to process each patch, rather than the 5–10 seconds that are spent when each $\psi_t^{[i]}$ is cached. This limitation may be difficult to totally overcome, but the methods we will propose in the concluding chapter for improving the importance sampler's efficiency may help.

The second limitation is that this use of the information from the first frame is only valid when the non-translational components of the patch's state at time t are small. When this is not the case, the displacement that each $\psi_t^{[i]}$ assigns minimum energy to may still be close to correct (if the rotational component is not too large), but these minimum energy displacements will not be uniform over all $\psi_t^{[i]}$, and so multiplicatively combining the local likelihoods may not lead to an importance sampling distribution that peaks close to the correct state. The implementation of solutions to this problem is one of the areas we intend to focus on in future. For now, we will briefly discuss some ideas that may lead to solutions.

In order to solve this limitation, we need to make an adjustment to our current Product-of-Local-Likelihoods model, which relies on the assumption that the time $t - 1$ particle states are not far from the true time t patch state. The simplest adjustment to make would be to still use the first image, but to replace the single Gibbs sampling iteration that we use to approximately sample from ℓ_t (as discussed in section 3.3.3) with multiple iterations. Unfortunately, it is not clear how many iterations would be needed to achieve an adequate degree of convergence.

Another alternative is to redefine the distributions that we sample each transformation component from in a way that breaks the cycles of dependencies between them, so that sampling

a state for the n^{th} component does not change the information that we used to define the distributions that we sampled the first $n - 1$ components from. Distributions like this only permit information to “flow forwards”, hence rendering multiple Gibbs sampling iterations unnecessary.

One way to create such distributions would be to define them in such a way that the probability they give to any given state of the current transformation component is a strictly-decreasing function of the uncertainty that remains over the states of the transformation components that are yet to be sampled. E.g. suppose we were given a particle state $\Theta_{t-1}^{[i]}$ and energy functions $\psi_t^{[i]}(\mathbf{d}; \mathbf{g})$ that use the first frame to define the likelihood energy of the event of a subregion centred at point \mathbf{g} undergoing a displacement \mathbf{d} . Furthermore, suppose we were only interested in estimating the orientation $\Theta_{\phi,t}^{[i]}$ and displacement $\Theta_{[d_x,d_y],t}^{[i]}$ of the particle at time t . Rather than sampling the particle’s displacement first and then sampling its orientation, suppose we started with an estimate ϕ of its orientation, and let $\Theta^{[i]}$ be an intermediate particle state with displacement $\Theta_{[d_x,d_y],t-1}^{[i]}$ and orientation ϕ . We could define a distribution f_D over the space D of displacements as normal, adding up the local likelihood energies of the transformation induced by $\Theta'^{[i]}$ when it is displaced by each $\mathbf{d} \in D$. So the uncertainty that would remain after assuming the particle’s orientation to be ϕ is given by f_D ’s entropy:

$$\rho(\phi) = - \sum_{\mathbf{d} \in D} f_D(\mathbf{d}) \ln f_D(\mathbf{d}) . \quad (6.59)$$

For an accurate choice of ϕ , f_D would be close to a Delta function (if the magnitude of any non-rigid transformation was small), and so $\rho(\phi)$ would be close to zero. Under an inaccurate choice of ϕ , there would be little agreement between the energy terms that each subregion contributes to any hypothesised displacement, and so $\rho(\phi)$ should be large. Hence, we could define an importance sampling distribution over $\Theta_t^{[i]}$ ’s orientation as a strictly-decreasing function of $\rho(\phi)$.

The problems with this approach however are that it would be computationally expensive to use when estimating more than two transformation components, and the distributions for all but the last sampled component would fail to preserve the exponential rate of growth in sampling

probability ratios w.r.t. the number of subregions, so they would have greater uncertainty. However this method could potentially be used to initialise the Product-of-Local-Likelihoods Gibbs sampler with an improved estimate of the particle's orientation, in the hope of reducing the number of iterations that are needed.

6.6 Results

A direct validation of the results of the methods we have discussed in this chapter for estimating the likelihood function parameters and the test threshold would involve comparing the results we obtained to those we would have obtained if we had based our construction of the D_* difference sets on the true deformations that the myocardiums underwent over thousands of frames. Manually tracking ~ 30 landmarks over thousands of frames for each patient in order to calculate this many deformation fields is an impractical task, so we are not in a position to carry out this form of validation. So instead, we will analyse the particle filter's performance under our estimates of the likelihood function parameters by using the deformation fields that we do have to validate the sampled particle states. In addition, we will attempt to validate the accuracy of our loss test (and hence our test threshold selection method) by comparing the errors in the states of the lost particles to the errors in the states of the particles that are not labelled as lost.

For each video sequence, we used this chapter's methods to estimate the parameters for patches initially centred at 14 of our manually placed landmarks, and we tracked them over all frames that the deformation fields were defined in, so that we could validate the results. We used the resampling approach to restore lost particles whenever a patch had at least one non-lost particle, and left them to drift otherwise. The random perturbation bounds and the other parameters we used to simulate the patch difference sets are summarised in table 6.1. Screenshots of the results are shown in figures 6.9 and 6.10, and graphs of the trajectories of the weighted means of the particles are shown in figure 6.11.

The fact that we only ever displace the subregions used in the calculation of the local likeli-

Parameter	Value		
	$D_{\ell'}$	D_L/D_F	D_B
Max. orientation perturbation	3°	5°	10°
Max. skew perturbation	0.0375	0.075	0.2
Max. scale perturbation	1.0375	1.075	1.25
Max. displacement perturbation	3p	5p	$\sqrt{2 \times 40^2}p = 56.6p$
No. deformation trajectories	5	10	10
No. frames per deformation trajectory	50	50	50
No. perturbations per frame	1	10	180

Table 6.1: This table summarises the parameters that we used to define the distributions of the perturbations that we applied when constructing the D_* set of patch/subregion differences. As always, the unit p stands for pixels.

hoods means that the dissimilarity measures that we calculate between pairs of subregions are always based on the comparison of pixels that do not quite correspond to each other, so it was unnecessary for the perturbations used in the construction of $D_{\ell'}$ to be as large as those used in the construction of D_L/D_F . Furthermore, since each patch contains multiple subregions (we gave percentiles of the distribution of the number subregions per patch in our earlier tests in table 3.2 on page 72), we did not need to apply multiple perturbations per frame when constructing $D_{\ell'}$. We used many more perturbations per frame in the construction of D_B than in the construction of D_L/D_F because the range of perturbations for D_B was bounded by the importance sampler’s range of transformations, and hence was much larger. This range of transformations and the other particle filter parameters that we used were exactly the same as those listed in table 3.1 on page 72, with the exception that we changed the outlier cutoff p for the \mathcal{E} dissimilarity function used in the evaluation of the patch likelihood functions L_t from 0.75 to 0.99. The value of this parameter did not matter in that chapter, but for the analysis we are about to present, we found that the greater degree of discrimination given by a larger cutoff led to better results.

For each initial patch state Θ_0 , we used the deformation fields to calculate estimates $\Theta'_{0:T}$ of the patch’s trajectory, using the method we described in section 6.4.2. We then calculated the RSK decomposition of the estimated state at each time step, which allowed us measure the errors in each individual component of the trajectories of each particle $\Theta_{0:T}^{[i]}$.

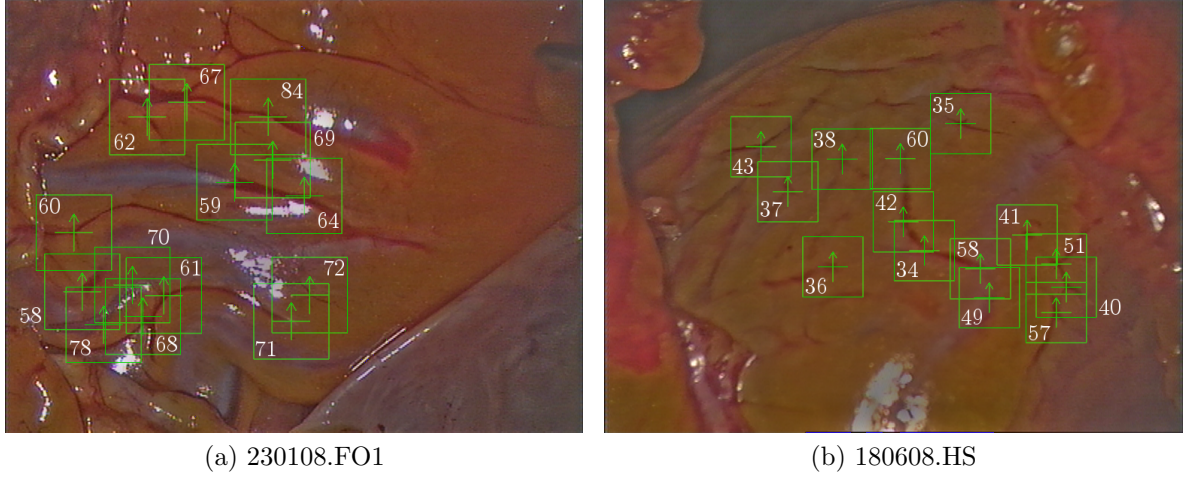


Figure 6.8: These figures show the ID numbers that we used to refer to each of the patches that we tracked.

In order for us to be able to compare the errors across different components, we calculated Euclidean errors between vertices, rather than directly calculating the errors in the state variable components. More specifically, suppose, for example, that we wanted to calculate the rotational error in particle $\Theta_t^{[i]}$. First of all we defined a transformation state $\hat{\Theta}$ by setting $\hat{\Theta}_{[\phi]} = \Theta_{t,[\phi]}^{[i]}$ and all other components to the corresponding components of Θ'_t , and then we considered the set of Euclidean errors between the four vertices of $\mathbf{V}(\Theta'_t)$ and those of $\mathbf{V}(\hat{\Theta})$, and also between the estimated patch centres $\mathbf{d}(\Theta'_t)$ and $\mathbf{d}(\hat{\Theta})$. We calculated the skewing and scaling component particle errors similarly. However for the displacement component, we calculated the Euclidean error between the particle centre $\mathbf{d}(\Theta_t^{[i]})$ and the time t position of the manually-tracked landmark, which is a more accurate reference than the deformation field. In addition, we also calculated the overall particle error as the set of errors between $\mathbf{V}(\Theta_t^{[i]})$'s vertices and $\mathbf{V}(\Theta'_t)$'s, and between $\mathbf{d}(\Theta_t^{[i]})$ and $\mathbf{d}(\Theta'_t)$.

6.6.1 Loss Test Results

Let $E_t^{[i]}$ be the set of overall errors for the time t state of the i^{th} of the 50 particles we used for a particular video. To test how well the loss test performed, we analysed the E sets *per frame* and *per patch*.

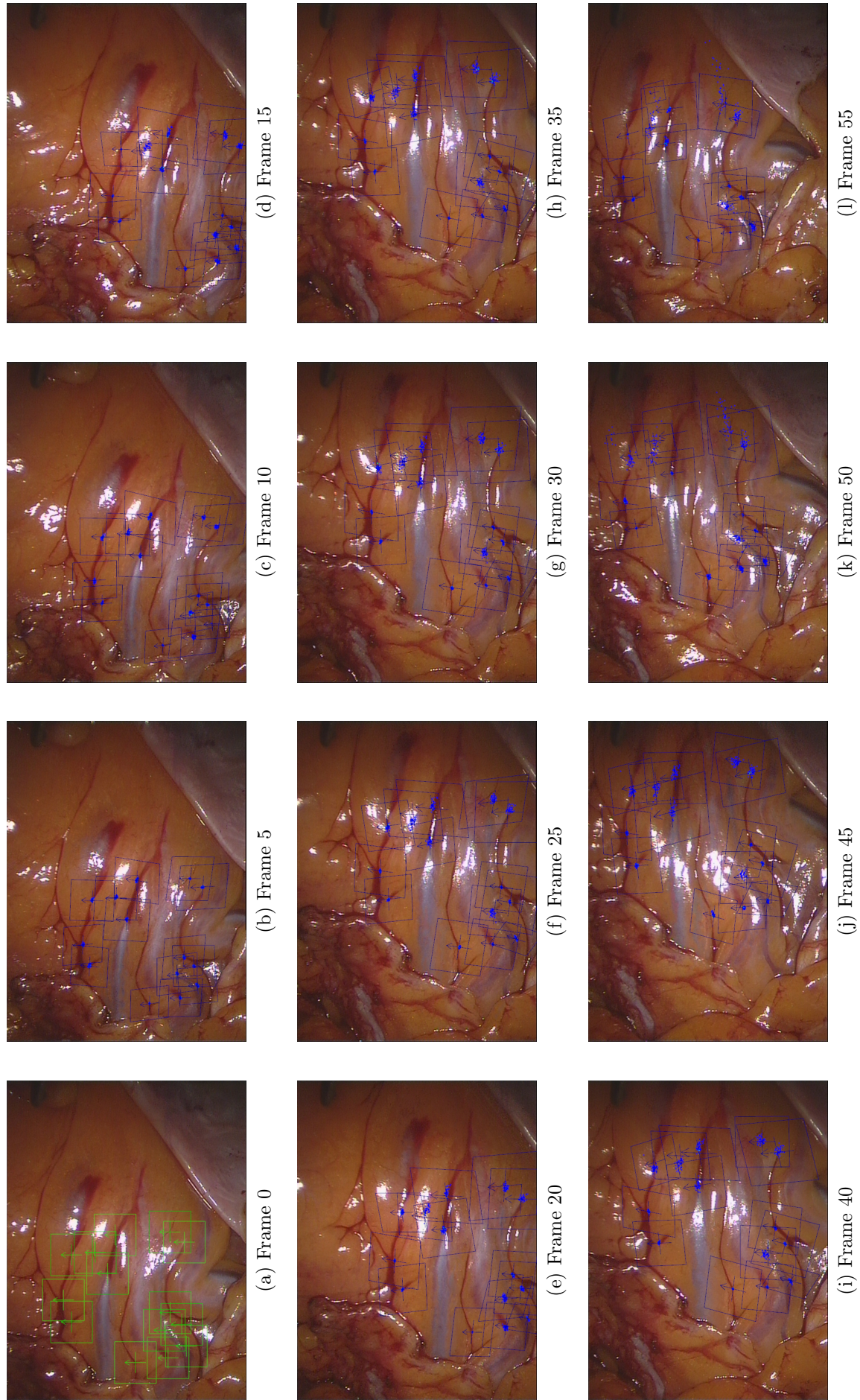


Figure 6.9: Screenshots of our particle filter's performance over the 230108.FO1 video, taken at 5 frame intervals. The quadrilaterals show the weighted means of the particles, and the mostly-blue crosses near their centres show the positions of the particles.

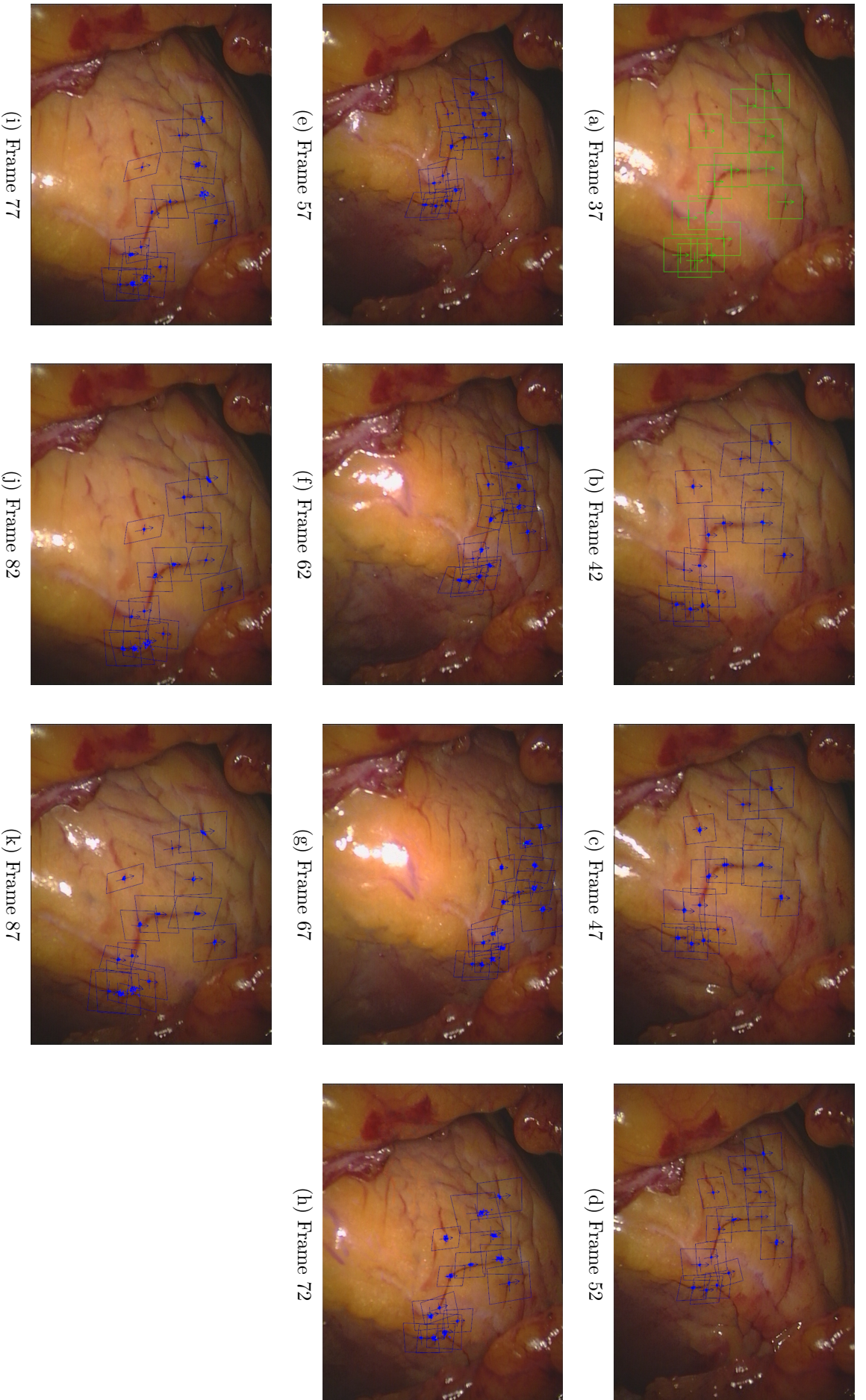


Figure 6.10: Screenshots of our particle filter's performance over the 180608.HS video, taken at 5 frame intervals. The quadrilaterals show the positions of the particles, and the mostly-blue crosses near their centres show the weighted means of the particles.

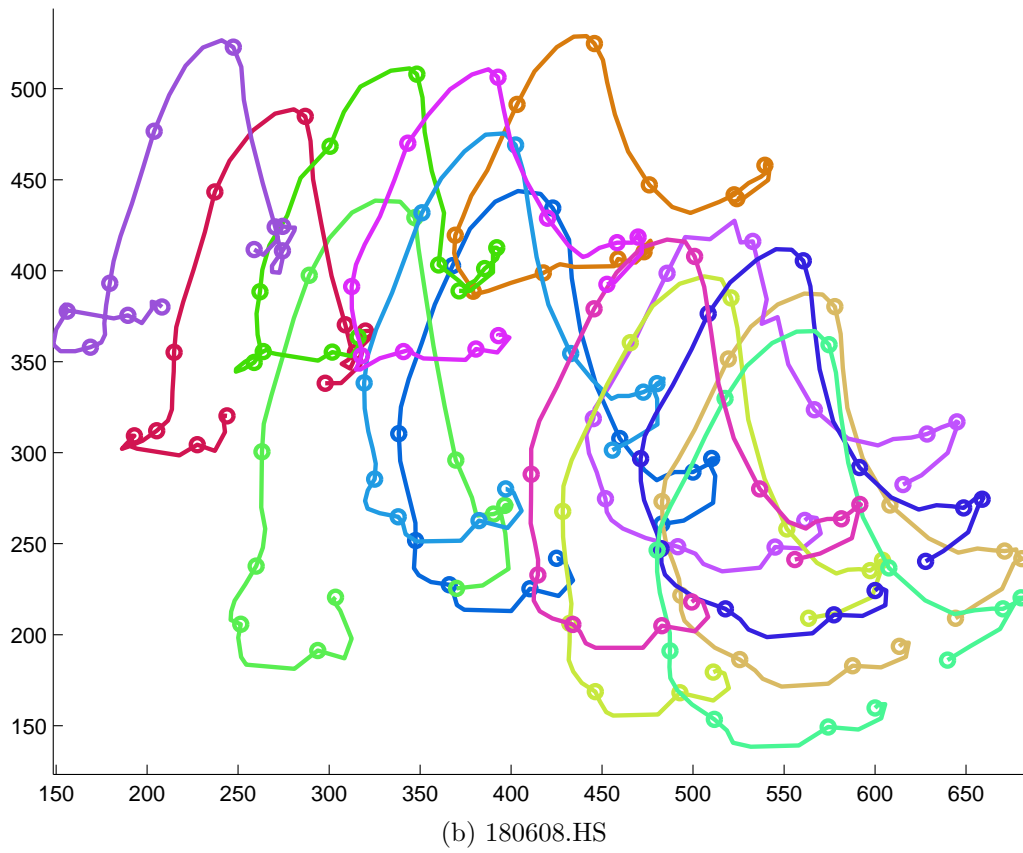
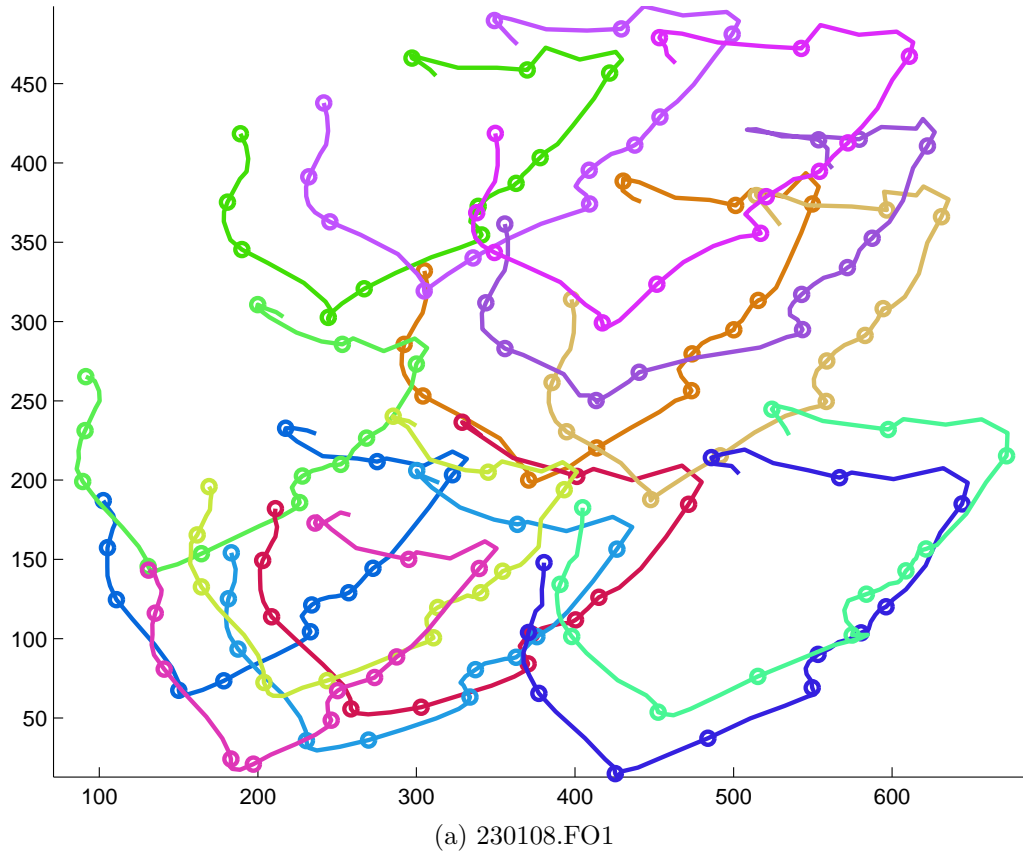


Figure 6.11: These graphs show the trajectories of the weighted means of the particles in each video. The circular markers denote the positions of the weighted means at 5-frame intervals.

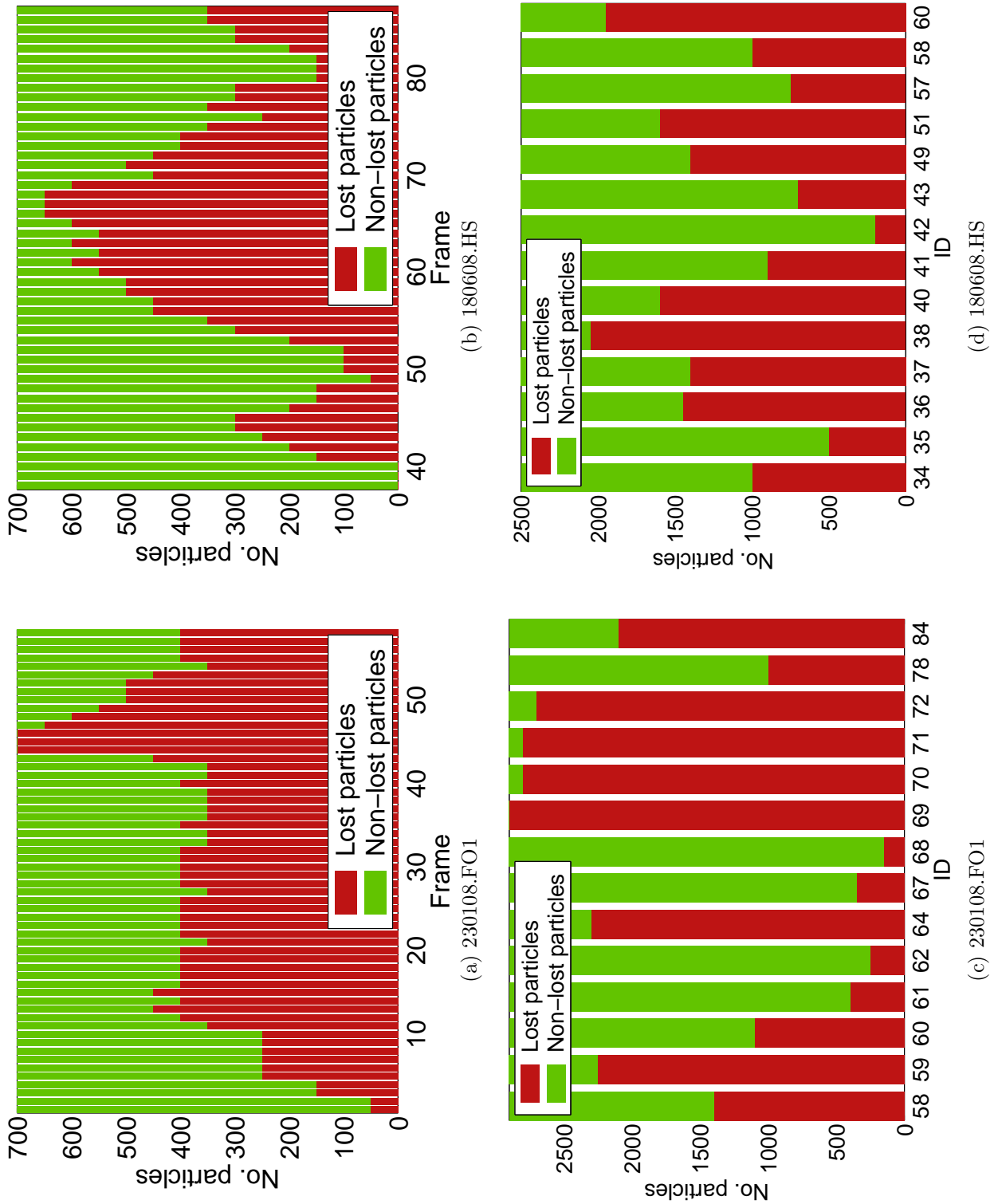


Figure 6.12: (a), (b) These figures show the changes in the distribution of the number of lost and non-lost particles over time. The numbers are always multiples of 50, which is the number of particles we used. This is because we resampled the lost particles whenever possible, meaning that either all of a patch's particles were lost after processing each frame, or none were. (c), (d) These figures show the numbers of lost and non-lost particles per patch, accumulated over all frames that they were tracked over. Again, the values are all multiples of 50.

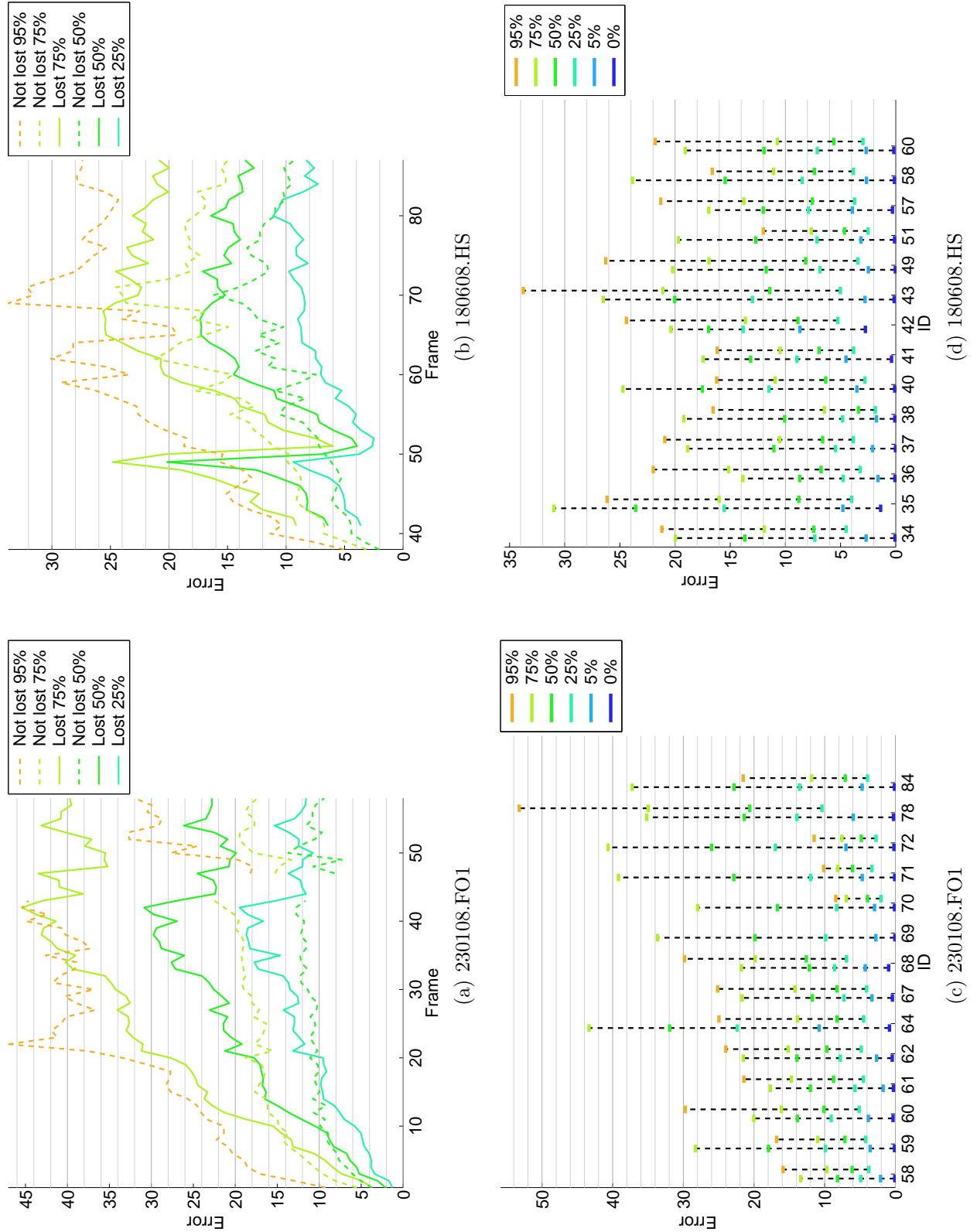


Figure 6.13: (a), (b) Error percentiles per frame after partitioning particles according to whether or not they were lost. (c), (d) Error percentiles per patch after partitioning particles according to whether or not they were lost. A pair of dashed vertical lines is given for each patch. The lines on the left mark the percentiles for lost particles, and the lines on the right mark the percentiles for non-lost particles. Patch 69 in 230108.FO1 only has one vertical line, because the loss test erroneously labelled all of its particles as lost.

For the per frame analysis, we constructed sets

$$E_{t,\top} \supseteq E_t^{[i]} \leftrightarrow W_t^{[i]} = \top \quad (6.60)$$

of errors associated with particles that were considered to be lost at each time t , and sets

$$E_{t,\perp} \supseteq E_t^{[i]} \leftrightarrow W_t^{[i]} = \perp \quad (6.61)$$

of errors for particles that were not considered to be lost at time t . Under an ideal loss test, we would expect the overall errors of the lost particles to be larger than those of the non-lost particles, so we have plotted percentiles of each of these sets as functions of t , to see the extent to which this was true. The results are shown in the top two graphs of figure 6.13, and the top two graphs of figure 6.12 show the total number of particles that were/were not lost in each frame.

For 230108.FO1, we see that for almost all time steps between frame 20 frame 59 (the final frame), 75% of the error terms from lost particles (i.e. the errors above the 25th percentile line) were greater than 50% of the error terms from non-lost particles. Over the same range of frames, we also see that 50% of the lost particle errors were greater than 75% of the non-lost particle errors. Note also that the numbers of lost and non-lost particles were roughly equal over 75% of these frames. There was less difference between the lost and non-lost particle errors over the first 20 frames, but there were also fewer lost particles over these frames.

A large proportion of particles were labelled as lost around the middle of the 180608.HS sequence. This was a result of the large degree of myocardial deformation and the large change in surface inclination around the middle of the sequence (as shown in figure 5.3 on page 123), which demonstrates the limitations of our reliance on the initial image as a reference for our loss test. Over most of the sequence, 50% of the lost particles had errors greater than 50% of the non-lost particles.

For the per patch analysis, we constructed a set

$$E_{\Theta_0, \top} \supseteq E_t^{[i]} \leftrightarrow \Theta_0^{[i]} = \Theta_0 \wedge \mathbf{W}_t^{[i]} = \top \quad (6.62)$$

for each initial patch state Θ_0 , made up of the overall errors associated with the patch's lost particles, and a set

$$E_{\Theta_0, \perp} \supseteq E_t^{[i]} \leftrightarrow \Theta_0^{[i]} = \Theta_0 \wedge \mathbf{W}_t^{[i]} = \perp \quad (6.63)$$

of the errors associated with its non-lost particles. We have plotted some of the percentiles of each of these sets in the bottom two graphs of figure 6.13, along with the number of lost/non-lost particles per feature in the bottom two graphs of figure 6.12. For the 230108.FO1 sequence, most of the patches either had far more lost particles than non-lost particles, or vice versa. This means that either the lost particle errors were much more densely sampled than the non-lost particle errors, or vice versa, and so it is not fair to compare the percentiles for these patches. For the patches that had a more even distribution of lost/non-lost particles, namely patches 58, 60 and 78, we see that the percentage of lost particle errors that exceeded 50% of the non-lost particle errors varies between 50% and a little under 75%, which is not too far from the per frame results above.

For 180608.HS, a larger number of patches had more balanced distributions of lost/non-lost particles. Of these patches, patch 40 had the best separation between lost particle errors and non-lost particle errors, with 75% of the lost particle errors being greater than 75% of the non-lost particle errors, and patches 41, 51 and 58 were next in line with 75% of the lost particle errors being ~ 2 or more pixels greater than 50% of the non-lost particle errors. For the remainder (i.e. patches 34, 36, 37, 49, 57), the percentages of lost particle errors greater than 50% of the non-lost particle errors were somewhere between 50% and 75%.

On the face of it, these figures seem to suggest that there is much room for improvement in the loss test's classification accuracy. However a significant limitation of this analysis is that it does not take into account the ambiguities in the states of some features; some of the patches, such as patch 78 in 230108.FO1 or patch 49 in 180608.HS, could slide around without their

appearance changing very noticeably. In the absence of strong additional (prior) information, it is not possible for the loss test or the posterior distribution to discriminate between these ambiguous states.

6.6.2 Particle Filter Accuracy

While the previous section's figures give a suggestion of how well the loss test performed, they are not indicative of the particle filter's accuracy, as they do not take the particle weights into account. So for each type of error that we measured (the overall errors, and the errors for each transformation component type), we constructed pairs $(E_t^{[i]}, w_t^{[i]})$ of particle error sets and particle weights, gathered these pairs together per patch over all frames, and then plotted percentiles of the resulting weighted distributions.

As shown at the top of figure 6.14, for 8 of 230108.FO1's 14 patches (those with IDs 58, 59, 60, 61, 62, 67, 68 and 70), the 75th percentiles of the overall weighted errors lay between 10 and 20 pixels, and for 13 of 180608.HS's 14 patches (all except for patch 51), the 75th overall error percentiles lay between 13 and 20 pixels. The displacement errors for these patches were generally smaller than this, with the first 6 of the 230108.FO1 patches listed above having 75th percentiles within about 8 pixels, and 11 of the 180608.HS patches mentioned above (all except for patches 41 and 42) also having 75th percentiles within about 8 pixels. The rotation results were similar: the 75th percentiles for 6 of the 230108.FO1 patches listed above were within 8 pixels, and the same percentiles for 10 of the 180608.HS patches mentioned above were also within about 8 pixels. The scaling results were not as good however. Only patch 58 in 230108.FO1 had a 75th percentile close to 8 pixels; the 75th percentiles of the other 7 patches listed above were between 10 and 14 pixels. For 180608.HS, only 3 patches had 75th percentiles less than 8 pixels; the 75th percentiles of the other 10 were between 10 and 21 pixels. The skewing results were better than the scaling results, but not as good as the displacement and rotation results: 2 of the 8 230108.FO1 patches listed above had 75th percentiles within 8 pixels, while the 75th percentiles of the remainder were between 9 and 13 pixels; 5 of the 13 mentioned 180608.HS patches had 75th percentiles that were less than 8 pixels, while 7 of the

remaining 75th percentiles lay within 9 and 14 pixels.

The overall errors and displacement errors for patch 51 in 180608.HS were worse than those of the other patches due to the patch being occluded by some loose tissue on the right-hand side towards end of the sequence (see frames 77–87 of the screenshots in figure 6.10). The 6 patches of 230108.FO1 that we did not list above (patches 64, 69, 71, 72, 78 and 84) performed worse for similar reasons, e.g. specular highlights occluded some of the textural details that patch 69 was tracking, and some of patch 71’s textural detail was lost when it got too close to the image boundary. Despite the overall errors for these patches being worse, it is interesting to note that some of their transformation component errors were small, presumably as a result of the smoothness of the myocardium’s deformations. E.g. the 75th percentile of patch 51’s skewing errors in 180608.HS was about 4 pixels, and the same percentile of patch 64’s rotation errors in 230108.FO1 was about 6 pixels.

6.6.3 Comparative Tests

We performed two comparative tests with each video: one to see the effects of only estimating the displacements of the patches, rather than full affine transformations; another to compare the performance of our particle filter to that of a standard tracking algorithm from the literature. For the former comparison, we carried out the parameter estimation procedures that we have described in this chapter as normal, except that we set all non-translational transformation components to the identity transformation, so that the transformations involved in the construction of the D_* image difference sets were all displacements. For the latter comparison, we used the OpenCV implementation of the pyramidal KLT tracker (p-KLT) described in [14] (with 3 pyramid levels), which augments the standard KLT tracker [66, 112, 99] with a coarse-to-fine scheme, so as to allow for the estimation of large displacements.

It is interesting to compare the particle filter’s performance when estimating full affine transformations to its performance when only estimating displacements for two reasons. Firstly, it is interesting to see how well our particle filter copes under the simplest useful transformation model. Secondly, since the space of displacements is small enough for our importance sampler

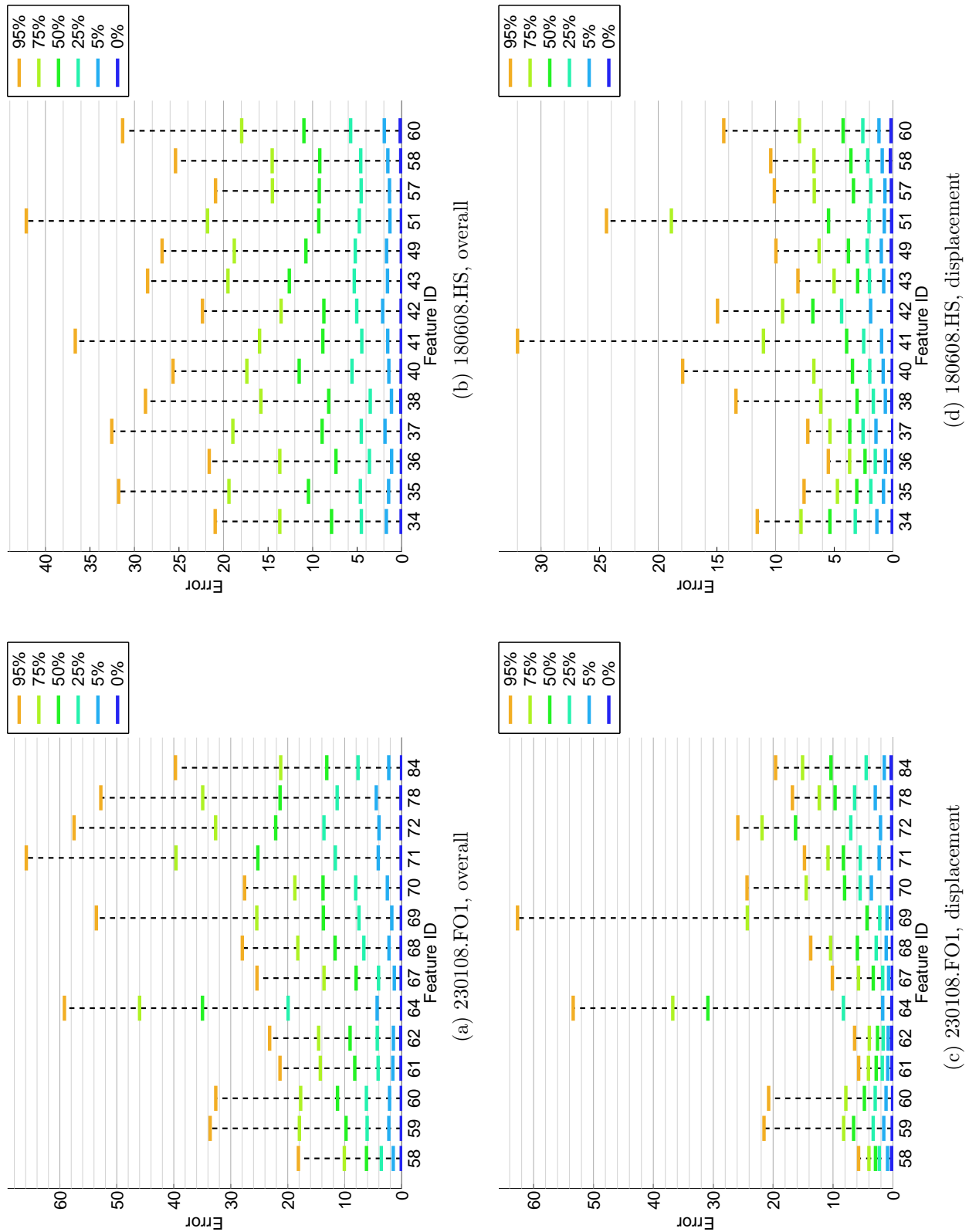


Figure 6.14: These figures show weighted percentiles of the overall patch errors and displacement patch errors.

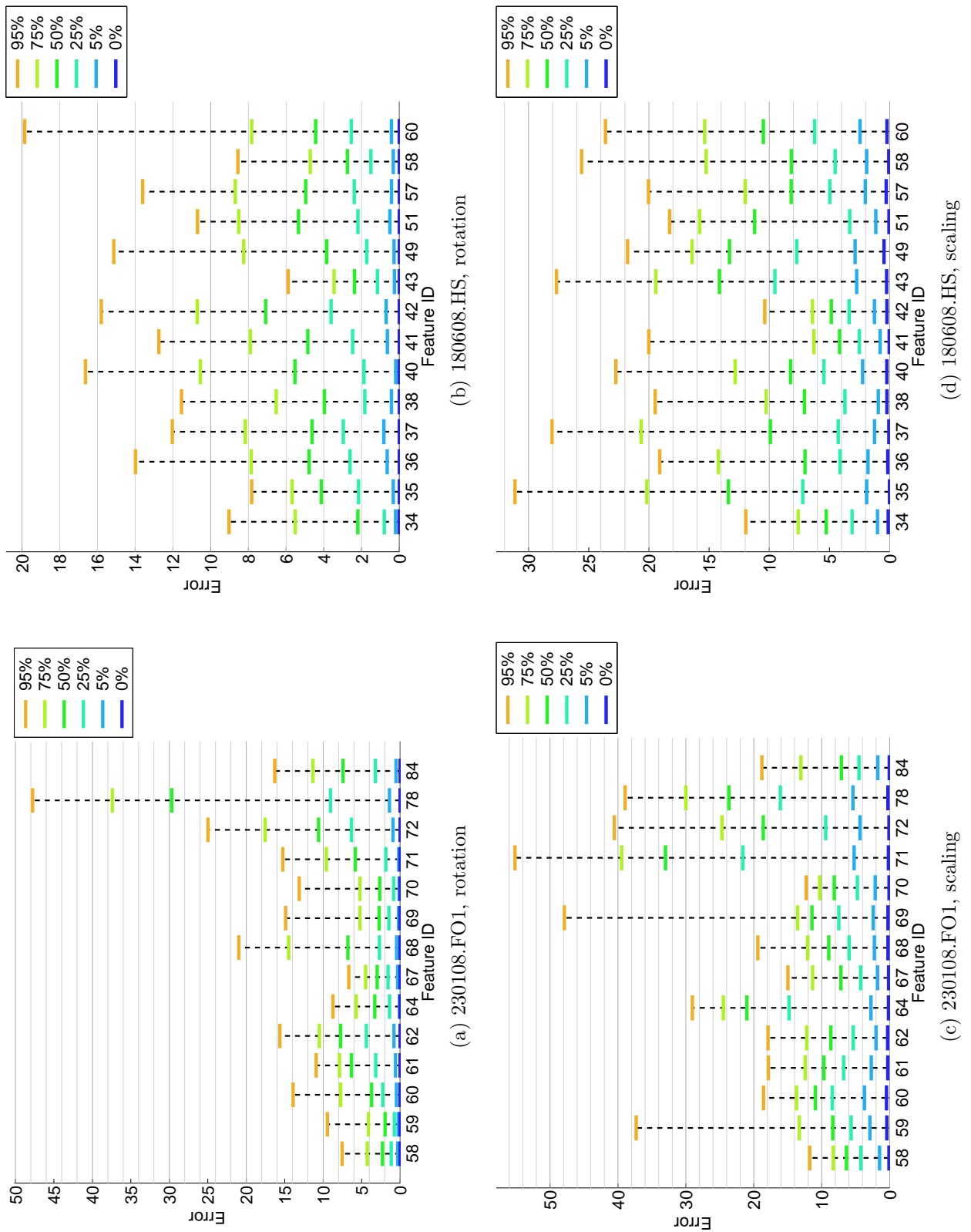
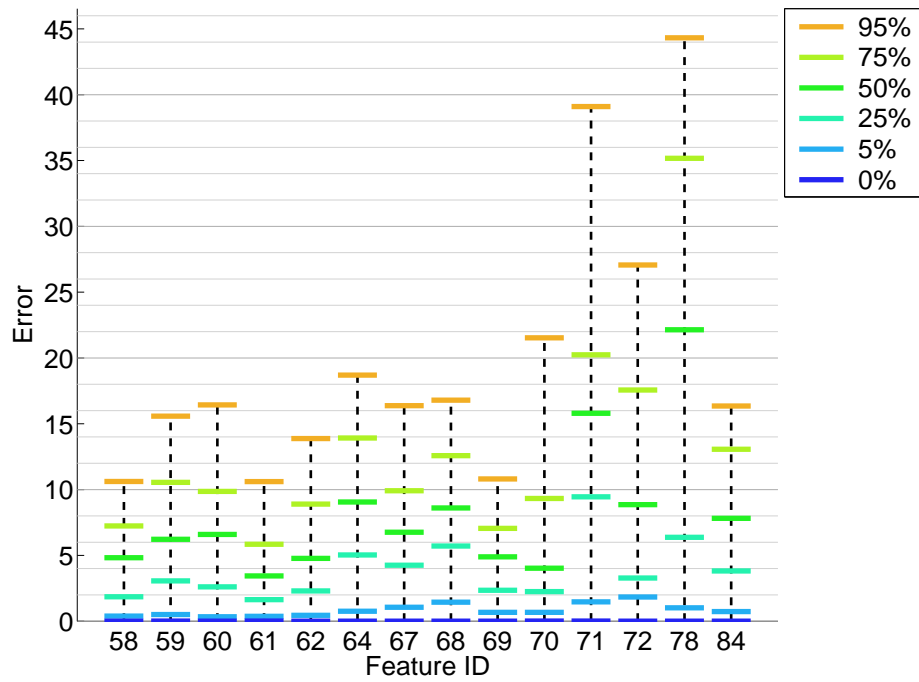
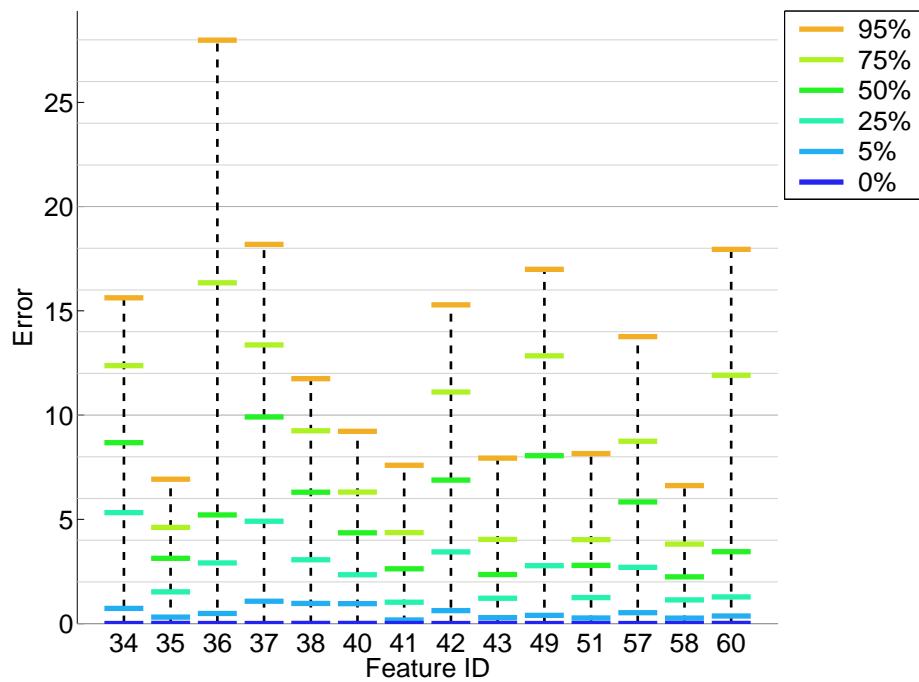


Figure 6.15: These figures show weighted percentiles of the rotational and scaling patch errors.



(a) 230108.FO1, skewing



(b) 180608.HS, skewing

Figure 6.16: These figures show weighted percentiles of the skewing patch errors.

to be able to sample from it directly, without having to resort to Gibbs sampling (as it does when sampling affine transformations), it is interesting to see whether or not the errors that result from the fact that we only carry out a single Gibbs iteration have an adverse effect on the estimation of the displacement component when using full affine transformations.

Figure 6.17 shows the weighted displacement error distributions for the first test. Most of the percentiles we have plotted differ by no more than 2 or 3 pixels, suggesting that a single Gibbs iteration was usually sufficient to estimate displacements when using affine transformations, and also that in most cases our particle filter was able to cope with the appearance changes caused by non-translational deformations when we were only estimating displacements. Interestingly, the patches of 230108.FO1 for which there are large differences between the two tests are mostly the 6 patches that had large overall errors in the results that we presented in the previous section. With the exception of the 95th percentile of patch 69, the error percentiles for patches 64, 69, 72 and 84 were smaller when we tracked full affine transformations, but the error percentiles for patches 71 and 78 were smaller when we only tracked displacements. The 75th and 95th percentiles for patch 51 in 180608.HS (which had the largest overall errors in that video at the 95th percentile) were also much larger when we tracked full affine transformations, as was the 95th percentile of patch 41.

Figure 6.18 shows the results of our comparison of our particle filter with the p-KLT tracker. For each frame of the videos, the results show percentiles of the distribution of displacement errors over all patches, measured with reference to the manually tracked landmarks (for our particle filter, we compared the landmarks to the weighted mean particle displacements). We ran the p-KLT tracker in two different ways: in one run, we used the first frame as its reference image and centred its reference patches over the landmarks in that frame, and in the other, we used the previous frame as its reference image and centred the reference patches over its estimates of the patch positions in that frame. In both cases, we took the most recently estimated patch positions as the initial estimates of their new positions. To make the comparison fair, we set the patches that we used with the p-KLT tracker to roughly the same size as the patches that we used with our particle filter (they could not be exactly the same size, as the p-KLT tracker required the dimensions to be odd, so we used 101×101 patches for 230108.FO1 and 81×81

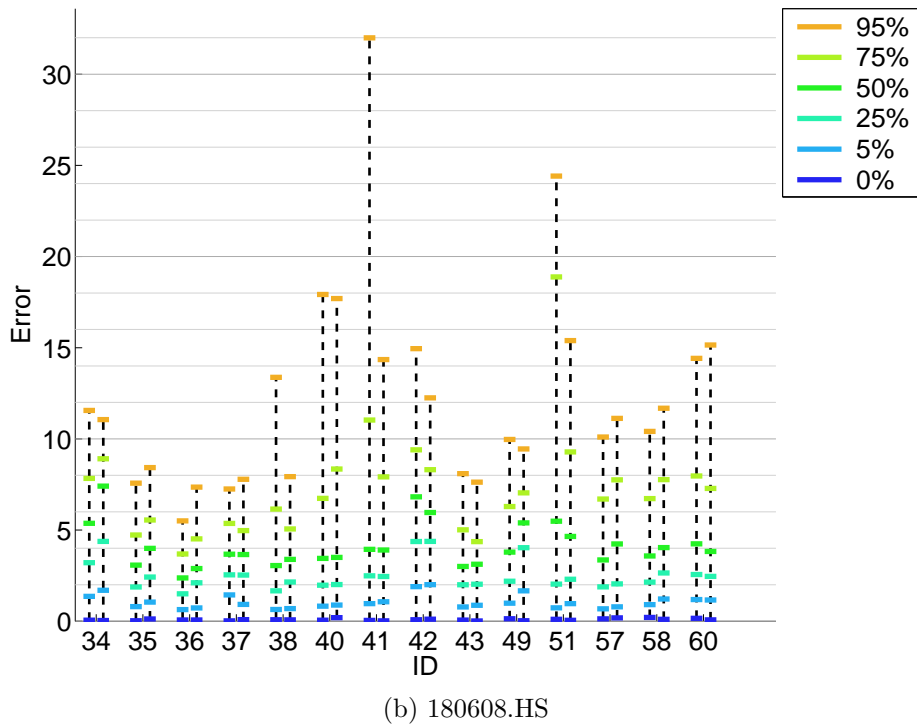
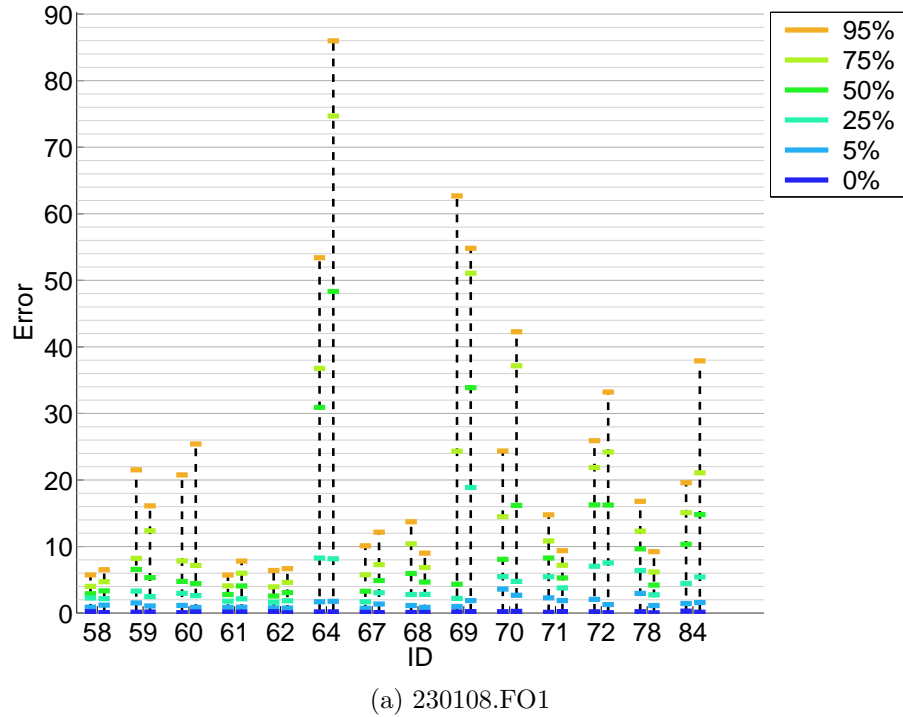


Figure 6.17: These figures show some percentiles of the weighted displacement error distributions for each patch in the tests in which we compared the particle filter’s performance when estimating full affine transformations to its performance when only estimating patch displacement. The left-hand column of each patch’s results gives the error percentiles from the tests in which we estimated full affine transformations, and the right-hand column gives the displacement-only percentiles.

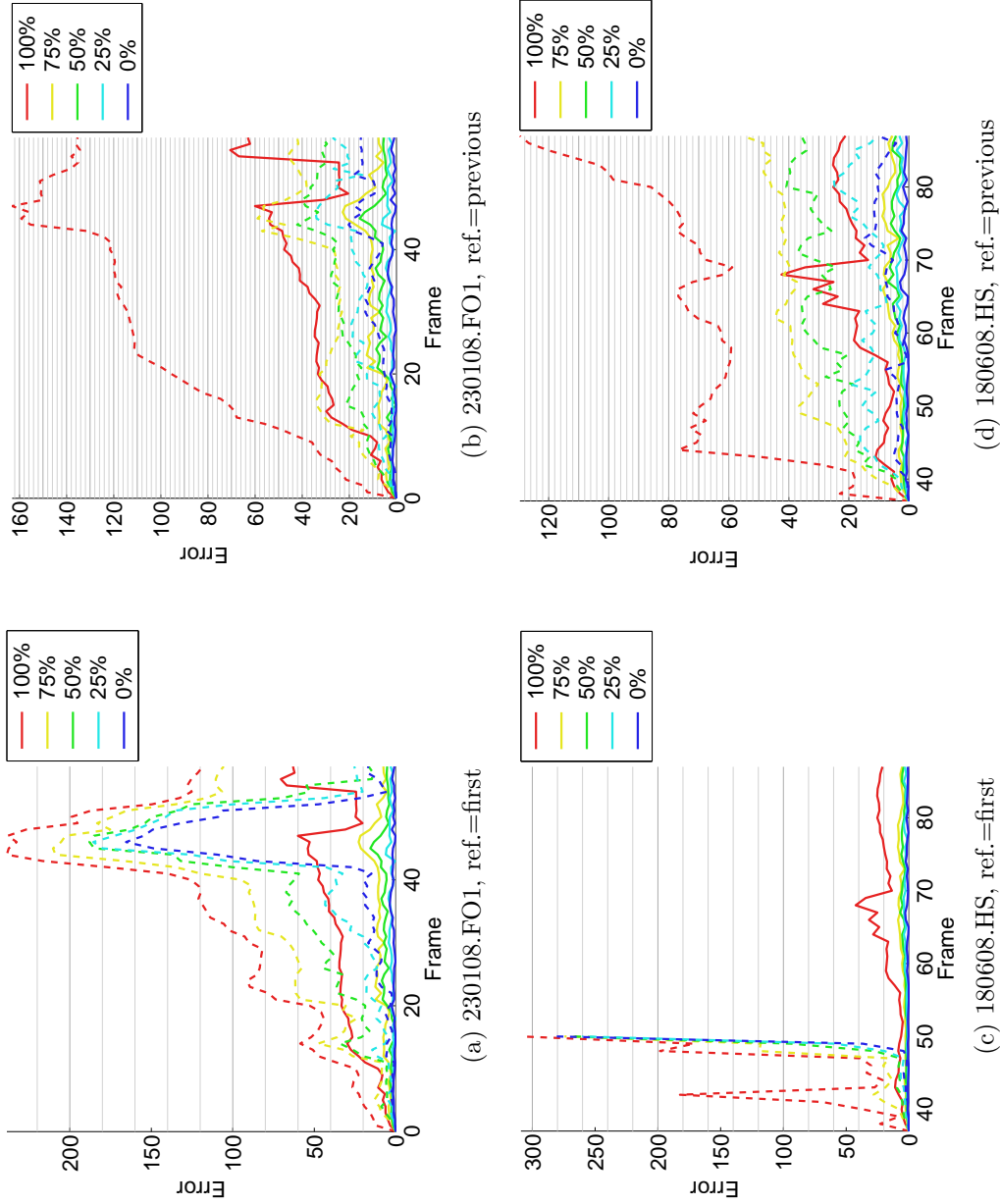


Figure 6.18: For each frame of the two videos, these figures show the percentiles of the distributions over the 14 patches of the pyramidal KLT tracker's (p-KLT, [14]) displacement errors and the errors in our particle filter's weighted mean estimates of the patch displacements. The dashed lines show the p-KLT tracker's results, and the solid lines show our particle filter's results. In the figures labelled "ref.=first", we used the first frame as the p-KLT tracker's reference image and centred its reference patches over the positions of the landmarks in that frame, and in the figures labelled "ref.=previous", we used the previous frame as the reference image, and centred its reference patches over its estimates of the positions of the patches in that frame.

patches for 180608.HS, as opposed to the 100×100 and 80×80 patches that we used with our particle filter).

The p-KLT tracker generally performed better when we used the previous frame as a reference rather than the first frame. When using the first frame, it performed particularly poorly on the 180608.HS video, declaring that it had lost all of the patches within the first 15 frames. But even when using the previous frame, the 25th percentiles of its errors were larger than our particle filter's 75th error percentiles in almost every frame, meaning that over 75% of its errors were larger than 75% of our particle filter's errors most of the time.

6.7 Conclusion

In this chapter, we have described a way of estimating the parameters of our particle filter's likelihood functions by using the DTMs we described in the previous chapter to simulate deformations of the regions around each patch, from which we construct sample sets of patch/subregion differences that we pass to a maximum-likelihood parameter estimator. We have also described initial solutions to the problems of determining when particles are irretrievably lost (based on a background/foreground likelihood ratio test), and restoring lost particles (based on resampling from the set of particles that are not considered to be lost). Our tests so far seem to indicate that more work needs to be done to improve the classification accuracy of our particle loss test, e.g. by taking into account recent information about the appearance of the patch.

We have presented test results that suggest that our particle filter estimates the displacements and orientations of patches reasonably well, but that it struggles to estimate their scale and skew. It may well be that the patches do not contain enough information to estimate scale and skew accurately, so taking into account the dependencies between the patches seems like the most promising way of improving the estimation accuracy. Our comparison of our particle filter with the pyramidal KLT tracker ([14]) suggests that the latter is incapable of achieving the degree of displacement estimation accuracy over myocardial image sequences that we have achieved through the techniques laid out in this thesis.

Chapter 7

Conclusion

7.1 Summary of Thesis Achievements

In this thesis, we have described methods that we have designed and tested as part of our research into reliable approaches by which the motion of the myocardial surface can be estimated, for use in image-guided surgery. Our methods were built upon a Sequential-Importance-Sampling-based particle filtering framework (with generalised dependency assumptions) for tracking affine changes in the states of surface patches, with an emphasis on information that can be extracted from the images in the absence of explicit prior assumptions.

In chapter 3, we introduced a Product-of-Experts-inspired model of the likelihood component of a patch’s posterior distribution. We showed that the uncertainty of the event of a subregion undergoing a given transformation can be reduced by multiplicatively combining estimates of the likelihood of that transformation for each subregion (as evaluated by local likelihood functions), and that this idea could be combined with Gibbs sampling to define low-variance importance sampling distributions. We also described conditions under which it seems reasonable to assume an exponential rate of growth in the ratio between importance sampling probabilities, and we validated this assumption empirically. This allowed us to reformulate the likelihood product approach in a way that accounts for the common event of subregion likelihood terms being unobservable. The reformulation expressed the importance sampling energy as the product of

the mean energy of the observable likelihood terms and an order of magnitude term.

In chapter 4, we described the dissimilarity function that we use to define the likelihood functions from the previous chapter. We treated the problem of measuring the dissimilarity between two patches as one of summarising the weighted distribution of squared differences between them, and we proposed the use of the Earth Mover’s Distance between the bottom $100p\%$ of this distribution and the Dirac delta function as a summariser, with a degree of sensitivity to outliers controlled by p . To make the dissimilarity function invariant to appearance changes caused by changes of illumination, we devised a simple image filtering algorithm based on a variant of the Dichromatic Reflection Model, that uses parametric surfaces to separate local lightness variations caused by albedo variations from those caused by illumination geometry variations.

Our aim in chapter 5 was to develop low-dimensional, periodic, generative models of affine and non-affine myocardial deformations, to assist in our attempts to automatically parameterise the particle filter’s likelihood functions. We constructed training data sets by fitting Thin-Plate-Splines-based deformation fields to sequences of manually tracked landmarks, randomly sampling a set of quasi-uniformly distributed similar canonical grids that lay within the images of the deformation fields, deforming them with the fields and removing the affine components of the resulting grids. This allowed us to create PCA models of the main modes of non-affine grid deformation much like the well-known Point Distribution Model. We then used these models to form sequences of reduced-dimensionality representations of the non-affine grid deformations, to which we appended log-space representations of the affine transformations, and produced new PCA models of the distribution of periodic B-spline approximations of these sequences, which we refer to as Deformation Trajectory Models (DTMs).

Finally, we introduced an idea for testing whether or not a particle is “lost” (i.e. whether or not it has fallen into a region of low marginal posterior probability) by comparing “foreground” and “background” likelihood functions of its state in chapter 6. Using our DTMs, we were able to construct sets of patch/subregion differences, from which we could calculate maximum likelihood estimates of: the parameters of these likelihood functions, the parameters

of the importance sampler's local likelihood functions, and the test threshold. The background likelihood function was simply represented as a smoothed histogram, and we were able to use standard gradient-based optimisation methods to estimate the remaining likelihood function parameters under the assumption that these functions were stretched exponentials. The objective function for the test threshold was not differentiable everywhere however, and it usually had numerous local minima, so we developed a modified version of the dichotomous search algorithm to minimise it. In addition to this, we also described some of our initial attempts at restoring lost particles by resampling them or using information from the first frame.

7.2 Future Work

Improvements to the efficiency and accuracy of our particle filtering system must be made before it will be suitable for image-guidance in a clinical setting. The main bottleneck at present is the calculation of the local likelihood functions. All of the tests we have done in this thesis have involved evaluating local likelihood functions over 41×41 grids, with a median of about 50 subregions per patch, which means that the lower bound on the median number of dissimilarity function evaluations per patch is about $50 \times 41^2 = 84050$. Caching helps to prevent this lower bound from growing much larger when the particles are in close agreement. But when caching cannot be used, this lower bound grows by a factor of the number of particles, which was 50 in our tests; $50 \times 84050 = 4202500$ dissimilarity function evaluations.

We calculated the local likelihoods at every point in the grid for simplicity. However, we should be able to significantly reduce the amount of work by making use of the following observations:

- a) local likelihood functions are quite spatially smooth locally, due to the smoothness of appearance of most parts of the myocardial surface;
- b) local likelihood functions for similar-looking neighbouring subregions are often similar in appearance, up to a coordinate system translation;

- c) a time $t+1$ local likelihood function for a particular subregion often looks similar to a time t local likelihood function for a subregion in an approximately corresponding position.

It may thus be possible to achieve good estimates of the local likelihood functions by evaluating the dissimilarity function at a well-selected sparse set of points, and then predicting the likelihood values everywhere else. So one aspect of our future research will involve investigating effective methods for selecting the evaluation points and carrying out the predictions.

Another important observation is that all values of all local likelihood functions for all particles can be calculated independently of each other. So we may also be able to gain a significant speedup by exploiting GPGPU parallelisation and/or multicore CPUs.

While we were able to greatly reduce the effects of the aperture problem through the use of the Product-of-Experts framework, it did cause problems when we were tracking patches with little texture, and patches that lay on vessels that were locally nearly straight. It seems that the only way to resolve this issue will be to take into account the dependencies between all of the patches we are tracking, e.g. by extending our Product-of-Experts approach to operate over all patches simultaneously. This would be a non-trivial extension to our current approach however, as we cannot assume that a single global affine transformation will account for the motion of all patches. If we can rebuild our deformation mode models and DTMs over larger areas, it might be possible to use them to enforce prior constraints on the set of deformations that the patches can simultaneously undergo. But this will only lead to a statistically efficient solution if we can find a way to directly incorporate these constraints into the calculation of the importance sampling distributions.

We also intend to extend our loss test to incorporate information from recent images, and to develop effective particle restoration techniques for the case when all of a patch's particles are labelled as lost. If the methods we described in the previous chapter prove to be infeasible or unhelpful, another potentially interesting avenue to explore would be a statistical reformulation of one of the many feature detection algorithms that have been published in the computer vision literature.

Finally, before our methods can be applied to more general (i.e. non-surgical) image sequences, it will be necessary to incorporate some form of motion segmentation process into the calculation of the importance sampling distributions and the patch likelihood function. For the importance samplers, this could potentially be done implicitly by replacing the calculation of mean energies with an outlier-insensitive average, such as the median. A more explicit approach would be needed for the patch likelihood function however, so as to maintain its discriminative power.

Appendix A

Notational Conventions

Symbol		Page
$\mathbf{a}, \mathbf{b}, \boldsymbol{\alpha}, \boldsymbol{\beta}$, etc.	Column vectors	
\overrightarrow{ab}	The directional vector $\mathbf{b} - \mathbf{a}$	
$\mathbf{a} \times \mathbf{b}$	The 2D cross product	128
$\triangle abc$	The triangle with vertices \mathbf{a} , \mathbf{b} and \mathbf{c}	
$\square abcd$	The quadrilateral with vertices \mathbf{a} , \mathbf{b} , \mathbf{c} and \mathbf{d} , which form a loop when connected in that order	
$\mathbf{a}, \mathbf{b}, \mathbf{A}, \mathbf{B}, \boldsymbol{\Theta}, \boldsymbol{\Phi}$, etc.	Random vectors	
a, b, A, B, Θ, Φ , etc.	Random scalars and any other type of random variable except for random vectors	
ϵ	A random noise vector	
$\mathbf{a} \sim f$	\mathbf{a} is sampled from PDF/PMF f	
$E[\mathbf{a}]$	The expected value of \mathbf{a}	
$E_p[\mathbf{a}]$	The expected value of \mathbf{a} under PDF p , i.e. $\int \mathbf{a} p(\mathbf{a}) d\mathbf{a}$	
$\mathbf{a} \mid \mathbf{b}$	The random variable that results from conditioning \mathbf{a} on \mathbf{b}	
$\mathbf{A}, \mathbf{B}, \boldsymbol{\Theta}, \boldsymbol{\Phi}$, etc.	Matrices	
\mathbf{A}^T	The transpose of \mathbf{A}	
$\mathbf{A}^{\frac{1}{2}}$	The square root of \mathbf{A} , i.e. a matrix \mathbf{B} such that $\mathbf{B}\mathbf{B} = \mathbf{A}$	

$\mathbf{0}$	The zero vector or matrix	
$\mathbf{0}_{m \times n}$	The $m \times n$ zero matrix	
\mathbf{I}	The identity matrix	
\mathbf{I}_n	The $n \times n$ identity matrix	
$x_{i:j}$	Indices i to j of x , where x is any indexable object	
$i : j$, where $i > j$	An empty index list	
$[a, b, c, \dots]$	The list of items a, b, c, \dots	
$[a, b]$	The list of items a and b , or the closed interval $\{x : a \leq x \leq b\}$. We indicate whenever we mean the closed interval.	
$\mathbf{A}_{i:,} \mathbf{A}_{:,j}$	Row i and column j , respectively, of \mathbf{A}	
$\mathbf{A}_{i:j:,} \mathbf{A}_{:,k:l}$	Rows i to j and columns k to l , respectively, of \mathbf{A}	
$\ \mathbf{A}\ _F$	The Frobenius norm of \mathbf{A} , defined as $\ \mathbf{A}\ _F \triangleq \sqrt{\sum_{i,j} \mathbf{A}_{i,j}^2}$	
\emptyset	The empty set	
x^\uparrow, x^\downarrow	Upper and lower bounds, respectively, of x	
$\delta_{i,j}$	The Kronecker Delta function, defined as $\delta_{i,j} = 1 \Leftarrow i = j$, $\delta_{i,j} = 0 \Leftarrow i \neq j$	
$\mathcal{M}(\mathbf{p}; \theta_a, \theta_b)$	The affine coordinate transformation that transforms \mathbf{p} by the inverse of the transformation defined by θ_a and then trans- forms the result by the transformation defined by θ_b	53
$N(x; \boldsymbol{\mu}, \boldsymbol{\Sigma})$	The Gaussian distribution over x with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$	
∂X	The boundary of set X	
$\varphi(x; \alpha, \gamma)$	The unnormalised stretched exponential function of x with exponent α and inverse decay rate γ	53
$\mathbf{V}(\theta)$	The four vertices of the patch after application of the trans- formation defined by θ	42

Appendix B

The After Math

B.1 Calculating the Decomposition $A = RSK$

The QR decomposition is one of the standard matrix decompositions in linear algebra, in which a matrix A is decomposed into the product of an orthogonal matrix Q and an upper-triangular matrix U (this matrix is usually referred to as R , but we will call it U to avoid confusion with the rotation matrix R), i.e.

$$A = QU \quad , \quad (B.1)$$

where Q and U are square matrices satisfying

$$QQ^T = I, \quad U_{i,j} = 0 \Leftarrow i > j \quad . \quad (B.2)$$

Many numerical libraries with linear algebra support can calculate this decomposition, using methods such as Householder reflections, Givens rotations or the Gram-Schmidt process (see [40] for details). However for the 2×2 matrices we are dealing with, the decomposition can be calculated with simple closed-form expressions, minimising the overhead introduced by the

generality of numerical library functions designed to support matrices of arbitrary size:

$$\mathbf{Q} \equiv \begin{pmatrix} x & -y \\ y & x \end{pmatrix}, \quad \mathbf{U} \equiv \begin{pmatrix} a & b \\ 0 & c \end{pmatrix}, \quad (\text{B.3})$$

where

$$\begin{aligned} a &\triangleq \sqrt{\mathbf{A}_{1,1}^2 + \mathbf{A}_{2,1}^2}, & b &\triangleq \frac{\mathbf{A}_{1,1}\mathbf{A}_{1,2} + \mathbf{A}_{2,1}\mathbf{A}_{2,2}}{a}, & c &\triangleq \frac{|\mathbf{A}|}{a} \\ x &\triangleq \frac{\mathbf{A}_{1,1}}{a}, & y &\triangleq \frac{\mathbf{A}_{2,1}}{a}. \end{aligned} \quad (\text{B.4})$$

The columns of \mathbf{Q} are clearly orthogonal to each other, and the squared magnitude of each column is

$$\left(\frac{\mathbf{A}_{1,1}}{a}\right)^2 + \left(\frac{\mathbf{A}_{2,1}}{a}\right)^2 = \frac{\mathbf{A}_{1,1}^2 + \mathbf{A}_{2,1}^2}{\mathbf{A}_{1,1}^2 + \mathbf{A}_{2,1}^2} = 1, \quad (\text{B.5})$$

hence \mathbf{Q} is orthogonal. In fact $|\mathbf{Q}| = 1$ as well*, so \mathbf{Q} is the rotation matrix \mathbf{R} .

The correctness of \mathbf{U} can be checked by comparing \mathbf{U} to $\mathbf{Q}^{-1}\mathbf{A}$:

$$\mathbf{Q}^{-1}\mathbf{A} = \frac{1}{a} \begin{pmatrix} x & y \\ -y & x \end{pmatrix} \mathbf{A} = \frac{1}{\sqrt{\mathbf{A}_{1,1}^2 + \mathbf{A}_{2,1}^2}} \begin{pmatrix} \mathbf{A}_{1,1}^2 + \mathbf{A}_{2,1}^2 & \mathbf{A}_{1,1}\mathbf{A}_{1,2} + \mathbf{A}_{2,1}\mathbf{A}_{2,2} \\ 0 & \mathbf{A}_{1,1}\mathbf{A}_{2,2} - \mathbf{A}_{2,1}\mathbf{A}_{1,2} \end{pmatrix}. \quad (\text{B.6})$$

This clearly simplifies to the expressions given above for \mathbf{U} 's parameters, as required.

Finally, we must solve

$$\mathbf{S}\mathbf{K} = \mathbf{U}, \quad (\text{B.7})$$

where \mathbf{S} is a diagonal matrix and \mathbf{K} is upper unitriangular (i.e. it is upper triangular and the elements on its main diagonal are 1). The solution to this is trivial:

$$\begin{aligned} \mathbf{S} &= \text{diag}(a, c), \\ \mathbf{K} &= \mathbf{S}^{-1}\mathbf{U}. \end{aligned} \quad (\text{B.8})$$

*When using numerical libraries to calculate the decomposition, \mathbf{Q} may have a negative determinant, in which case its i^{th} column and the i^{th} row of \mathbf{U} must be scaled by -1, for arbitrary i .

B.2 The Existence and Realness of 2×2 Matrix Logarithms

[74] gives the only cases under which a 2×2 matrix has a real logarithm. In the case of our $\mathbf{A}(\theta)$ matrices (upon which we enforce the restriction $|\mathbf{A}(\theta)| > 0$), the conditions are that either:

1. $\mathbf{A}(\theta)$ has equal negative eigenvalues and is diagonalisable;
2. $\mathbf{A}(\theta)$ is positive-definite; or
3. $\mathbf{A}(\theta)$'s eigenvalues are both complex.

Condition 1 is true if and only if $\mathbf{A}(\theta)$ is a negative isotropic scaling matrix, i.e. if $\mathbf{A}(\theta) = -\alpha \mathbf{I}$, for some $\alpha > 0$.

Condition 2 is true if and only if $\text{tr}(\mathbf{A}(\theta)) > 0$ (since $|\mathbf{A}(\theta)| > 0$). Expanding eq. (3.2) and assuming $\theta_{[\kappa_y]} = 0$, it follows that this is equivalent to the condition

$$(\theta_{[s_x]} + \theta_{[s_y]}) \cos(\theta_{[\phi]}) + \theta_{[s_x]} \theta_{[k_x]} \sin(\theta_{[\phi]}) > 0 \quad (\text{B.9})$$

It is difficult to give an intuitive description of the geometric significance of this, but a special case of interest is that it is always true when $\theta_{[k_x]} = 0$ and $|\theta_{[\phi]}| < 0.5\pi$ radians.

Another way to think about the significance of $\text{tr}(\mathbf{A}(\theta)) > 0$ is to use the visualisation shown in figure B.1, in which

$$\mathbf{A} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad (\text{B.10})$$

is represented by a parallelogram that depicts its effect on a unit square with a vertex at the origin. The positive trace condition means that either the x ordinate of the bottom-right vertex of the parallelogram must be less negative than the y ordinate of the top-left vertex is positive, or that the y ordinate of the top-left vertex must be less negative than the x ordinate of the bottom-right vertex is positive.

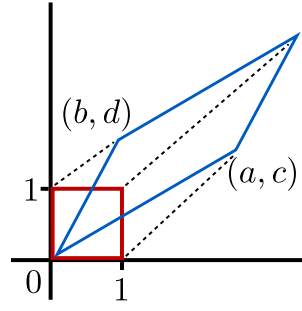


Figure B.1: The vertices of the red unit square are left-multiplied by a matrix \mathbf{A} defined as in the text, transforming the square into the blue parallelogram, which has area $|\mathbf{A}|$.

For condition 3, we use the form of \mathbf{A} given in figure B.1 again. By calculating \mathbf{A} 's characteristic polynomial and using the quadratic formula, it follows that \mathbf{A} 's eigenvalues λ satisfy

$$\lambda = \frac{a + d \pm \sqrt{(a + d)^2 - 4(ad - bc)}}{2} . \quad (\text{B.11})$$

Therefore, the eigenvalues are complex if and only if the discriminant is negative, i.e. if and only if

$$(a + d)^2 - 4(ad - bc) < 0 \Rightarrow \frac{|\text{tr}(\mathbf{A})|}{2} < \sqrt{|\mathbf{A}|} . \quad (\text{B.12})$$

In terms of figure B.1, the left-hand side of this inequality is the magnitude of the mean value of the x ordinate of the bottom-right vertex of the parallelogram and the y ordinate of its top-left vertex, and the right-hand side is the side length of a square that is equal to the parallelogram in area.

B.3 Calculating the Discretisation Resolution Parameters

In this section, we derive the parameters that control the discretisation resolution of the sets \hat{R} , \hat{K} and \hat{S} that we introduced in section 3.3.3. Our derivations are based on a patch $\mathbf{V}(\theta)^*$ with central point $\mathbf{d}(\theta)$. In all cases, the aim is to find a transformation parameter update δ_x for transformation type x (which can be any of the transformation parameter labels used in eq.

*Please see section 3.2 for a definition of this symbol and all other symbols that are not defined in this section.

(3.2)) such that the directional vector

$$\mathcal{M}(\mathbf{q}'; \theta, \theta^{+x}(\delta_x)) - \mathbf{q}' = \mathbf{A}(\theta^{+x}(\delta_x)) \mathbf{A}^{-1}(\theta)(\mathbf{q}' - \mathbf{d}(\theta)) + \mathbf{d}(\theta^{+x}(\delta_x)) - \mathbf{q}' \quad (\text{B.13})$$

is a unit vector (i.e. δ_x moves \mathbf{q}' one pixel), where \mathbf{q}' is the furthest point in $\mathbf{V}(\theta)$ from $\mathbf{d}(\theta)$, under some appropriate measure of distance, and $\theta^{+x}(\delta_x)$ is the result of updating $\theta_{[x]}$ by δ_x , but keeping θ 's other parameters the same. In particular, for rotation and skew:

$$\theta^{+x}(\delta_x) \triangleq \theta_{[x]} + \delta_x \quad , \quad (\text{B.14})$$

whilst for scaling:

$$\theta^{+x}(\delta_x) \triangleq \theta_{[x]} \delta_x \quad . \quad (\text{B.15})$$

Since we only need to calculate the discretisation parameters for rotation, skewing and scaling, we will always have $\mathbf{d}(\theta) = \mathbf{d}(\theta^{+x}(\delta_x))$. So if we define

$$\mathbf{q} \triangleq \mathbf{q}' - \mathbf{d}(\theta) \quad , \quad (\text{B.16})$$

then our objective is to find δ_x such that

$$1 = \|\mathbf{A}(\theta^{+x}(\delta_x)) \mathbf{A}^{-1}(\theta) \mathbf{q} - \mathbf{q}\|^2 \quad . \quad (\text{B.17})$$

B.3.1 The Rotational Discretisation Parameter δ_ϕ

Let \mathbf{q}' be the furthest vertex in $\mathbf{V}(\theta)$ from $\mathbf{d}(\theta)$, measuring distances with the Euclidean norm. The choice of \mathbf{q}' is arbitrary when more than one vertex is at least as far from the centre as all the others.

The rotation update δ_ϕ that displaces \mathbf{q}' by 1 pixel satisfies:

$$1 = \|\mathbf{R}(\theta^{+\phi}(\delta_\phi)) \mathbf{R}^{-1}(\theta) \mathbf{q} - \mathbf{q}\|^2 \quad , \quad (\text{B.18})$$

since $\mathbf{S}(\theta + \phi(\delta_\phi)) = \mathbf{S}(\theta)$ and $\mathbf{K}(\theta + \phi(\delta_\phi)) = \mathbf{K}(\theta)$. The matrix $\mathbf{R}(\theta + \phi(\delta_\phi)) \mathbf{R}^{-1}(\theta)$ is simply an anticlockwise rotation by δ_ϕ , which we will denote by the rotation matrix \mathbf{R}' .

We then have

$$\begin{aligned} 1 &= \|\mathbf{R}'\mathbf{q} - \mathbf{q}\|^2 \\ &= (\mathbf{q}^T \mathbf{R}'^T - \mathbf{q}^T)(\mathbf{R}'\mathbf{q} - \mathbf{q}) \\ &= 2\mathbf{q}^T \mathbf{q} - 2\mathbf{q}^T \mathbf{R}'^T \mathbf{q} , \end{aligned} \tag{B.19}$$

since $\mathbf{R}'^T \mathbf{R}' = \mathbf{I}$.

Hence

$$\begin{aligned} \frac{1}{2} &= \mathbf{q}^T \mathbf{q} - \mathbf{q}^T \begin{pmatrix} \cos(\delta_\phi) & -\sin(\delta_\phi) \\ \sin(\delta_\phi) & \cos(\delta_\phi) \end{pmatrix} \mathbf{q} \\ &= \mathbf{q}^T \mathbf{q} - \mathbf{q}_x(\mathbf{q}_x \cos(\delta_\phi) - \mathbf{q}_y \sin(\delta_\phi)) - \mathbf{q}_y(\mathbf{q}_x \sin(\delta_\phi) + \mathbf{q}_y \cos(\delta_\phi)) \\ &= \mathbf{q}^T \mathbf{q} (1 - \cos(\delta_\phi)) \\ \Rightarrow \delta_\phi &= \arccos \left(1 - \frac{1}{2\mathbf{q}^T \mathbf{q}} \right) . \end{aligned} \tag{B.20}$$

B.3.2 The Skewing Discretisation Parameters δ_{κ_x}

The rotation matrix $\mathbf{R}(\theta) = \mathbf{R}(\theta + \kappa_x(\delta_{\kappa_x}))$ has no effect on the magnitude of the displacement induced on any point by the horizontal skew update parameter δ_{κ_x} , so we can omit it from \mathbf{A} . Since $\mathbf{K}(\theta)$ is the first factor of $\mathbf{A}(\theta)$ that is applied, let \mathbf{v} be a vertex of $\hat{\mathbf{V}}$ with maximal y ordinate magnitude, and define $\mathbf{q}' = \mathbf{S}(\theta)\mathbf{K}(\theta)\mathbf{v} + \mathbf{d}(\theta) \Rightarrow \mathbf{q} = \mathbf{S}(\theta)\mathbf{K}(\theta)\mathbf{v}$. The δ_{κ_x} that displaces \mathbf{q}' by one pixel satisfies

$$\begin{aligned} 1 &= \|\mathbf{S}(\theta)\mathbf{K}(\theta + \kappa_x(\delta_{\kappa_x}))\mathbf{K}^{-1}(\theta)\mathbf{S}^{-1}(\theta)\mathbf{q} - \mathbf{q}\|^2 \\ &= \|\mathbf{S}(\theta)(\mathbf{K}(\theta) + \mathbf{K}')\mathbf{v} - \mathbf{S}(\theta)\mathbf{K}(\theta)\mathbf{v}\|^2 \\ &= \|\mathbf{S}(\theta)\mathbf{K}'\mathbf{v}\|^2 , \end{aligned} \tag{B.21}$$

where

$$\mathbf{K}' = \begin{pmatrix} 0 & \delta_{\kappa_x} \\ 0 & 0 \end{pmatrix}, \quad (\text{B.22})$$

As the bottom row of \mathbf{K}' is zero, eq. (B.21) reduces to

$$\begin{aligned} 1 &= (\theta_{[s_x]} \delta_{\kappa_x} \mathbf{v}_y)^2 \\ \Rightarrow \delta_{\kappa_x} &= \frac{1}{\theta_{[s_x]} |\mathbf{v}_y|}, \end{aligned} \quad (\text{B.23})$$

since we require δ_{κ_x} to be positive, as stated in section 3.3.3.

By symmetry, if we define \mathbf{v}' as a vertex of $\hat{\mathbf{V}}$ with maximal x ordinate magnitude, then

$$\delta_{\kappa_y} = \frac{1}{\theta_{[s_y]} |\mathbf{v}'_x|}. \quad (\text{B.24})$$

B.3.3 The Scaling Discretisation Parameters δ_s .

As in the skewing case, the rotation matrix $\mathbf{R}(\theta) = \mathbf{R}(\theta^{+s_x}(\delta_{s_x}))$ has no effect on the magnitude of the displacement induced on any point by the horizontal scale update parameter δ_{s_x} , so we can omit it again. Defining \mathbf{q} as a vertex of $\mathbf{S}(\theta)\mathbf{K}(\theta)\hat{\mathbf{V}}$ with maximal x ordinate magnitude, and observing that $\mathbf{K}(\theta) = \mathbf{K}(\theta^{+s_x}(\delta_{s_x}))$ and that in the absence of rotation, δ_{s_x} will only affect the x component of \mathbf{q} , reduces eq. (B.17) to

$$\begin{aligned} 1 &= (\delta_{s_x} \mathbf{q}_x - \mathbf{q}_x)^2 \\ \Rightarrow \frac{1}{\mathbf{q}_x^2} &= (\delta_{s_x} - 1)^2 \\ \Rightarrow \delta_{s_x} &= 1 + \frac{1}{|\mathbf{q}_x|}, \end{aligned} \quad (\text{B.25})$$

since $\delta_{s_x} > 1$, as stated in section 3.3.3.

Exploiting symmetry once again, it follows that

$$\delta_{s_y} = 1 + \frac{1}{|\mathbf{q}''_y|}, \quad (\text{B.26})$$

where \mathbf{q}'' is a vertex of $\mathbf{S}(\theta)\mathbf{K}(\theta)\hat{\mathbf{V}}$ with maximal y ordinate magnitude.

B.4 Maximising the Importance Sampling Distribution Entropy

Given an unnormalised discrete distribution $f(x)$ with domain X , defined at $x \in X_d \subseteq X$ and undefined at $x \in X_u = X - X_d$, we want to find an energy $\psi_u \geq 0$ that maximises $k^{-1}\tilde{f}$'s entropy H , where:

$$\tilde{f}(x; \psi_u) \triangleq \begin{cases} f(x) & , x \in X_d \\ e^{-\psi_u} & , x \in X_u \end{cases} \quad (\text{B.27})$$

$$H \triangleq - \sum_{x \in X} k^{-1} \tilde{f}(x) \ln(k^{-1} \tilde{f}(x)) \quad (\text{B.28})$$

$$k \triangleq \sum_{x \in X} \tilde{f}(x) . \quad (\text{B.29})$$

We will begin by defining \tilde{f} 's unnormalised entropy, H' , as

$$H' \triangleq - \sum_{x \in X} \tilde{f}(x) \ln(\tilde{f}(x)) , \quad (\text{B.30})$$

and making use of the identity:

$$\begin{aligned} H &\equiv -k^{-1} \sum_{x \in X} \tilde{f}(x) \left(\ln(\tilde{f}(x)) - \ln(k) \right) \\ &\equiv -k^{-1} \sum_{x \in X} \tilde{f}(x) \ln(\tilde{f}(x)) + \ln(k) k^{-1} \sum_{x \in X} \tilde{f}(x) \\ &\equiv k^{-1} H' + \ln(k), \quad \text{by eqs. (B.29) and (B.30)} . \end{aligned} \quad (\text{B.31})$$

From this, it follows that the derivative of H w.r.t. ψ_u satisfies

$$\begin{aligned}
 \frac{dH}{d\psi_u} &= k^{-1} \frac{dH'}{d\psi_u} + \frac{dk^{-1}}{d\psi_u} H' + k^{-1} \frac{dk}{d\psi_u} \\
 &= k^{-1} \frac{dH'}{d\psi_u} - k^{-2} \frac{dk}{d\psi_u} H' + k^{-1} \frac{dk}{d\psi_u} \\
 &= k^{-1} \left(\frac{dH'}{d\psi_u} + \frac{dk}{d\psi_u} (1 - k^{-1} H') \right) .
 \end{aligned} \tag{B.32}$$

Differentiating H' w.r.t. ψ_u gives

$$\begin{aligned}
 \frac{dH'}{d\psi_u} &= - \sum_{x \in X} \left(\frac{d\tilde{f}(x)}{d\psi_u} \ln(\tilde{f}(x)) + \frac{d\tilde{f}(x)}{d\psi_u} \right) \\
 &= |X_u| (1 - \psi_u) e^{-\psi_u} ,
 \end{aligned} \tag{B.33}$$

and since

$$k = \sum_{x \in X_d} \tilde{f}(x) + |X_u| e^{-\psi_u} , \tag{B.34}$$

differentiating k w.r.t. ψ_u gives

$$\frac{dk}{d\psi_u} = -|X_u| e^{-\psi_u} . \tag{B.35}$$

The entropy is maximised when

$$\begin{aligned}
 \frac{dH}{d\psi_u} &= 0 \\
 \Rightarrow \frac{dk}{d\psi_u} (1 - k^{-1} H') &= - \frac{dH'}{d\psi_u} \\
 \Rightarrow -|X_u| (1 - k^{-1} H') e^{-\psi_u} &= -|X_u| (1 - \psi_u) e^{-\psi_u} .
 \end{aligned} \tag{B.36}$$

If we let

$$\begin{aligned}
 A &= \sum_{x \in X_d} \tilde{f}(x) \\
 B &= \sum_{x \in X_d} \tilde{f}(x) \ln(\tilde{f}(x)) ,
 \end{aligned} \tag{B.37}$$

then this becomes

$$\begin{aligned}
\frac{-B + |X_u|\psi_u e^{-\psi_u}}{A + |X_u|e^{-\psi_u}} &= \psi_u \\
\Rightarrow -B &= A\psi_u \\
\Rightarrow \psi_u &= -\frac{B}{A} .
\end{aligned} \tag{B.38}$$

Finally, we need to check what non-negative value of ψ_u maximises the entropy when $-\frac{B}{A}$ is negative. Since k is always positive, replacing the $=$ in eq. (B.36) with $<$ and working through eq. (B.38) with these new inequalities tells us that $\frac{dH}{d\psi_u} < 0 \Rightarrow \psi_u > -\frac{B}{A}$, which means that H is a strictly decreasing function of ψ_u for all $\psi_u > -\frac{B}{A}$. Hence in general, the entropy is maximised by

$$\psi_u = \max\left(0, -\frac{B}{A}\right) . \tag{B.39}$$

B.5 Maximising a Particle Set's Entropy

Let J_u be a set of particle indices such that for each $j \in J_d$, the time t patch likelihood $L_t(\Theta_t^{[j]})$ is undefined, and let J_d be the set of indices of particles with defined time t patch likelihoods. Assuming, without loss of generality, that L_t has a maximum value of 1*, we want to find a patch likelihood energy $\psi_u \geq 0$ for each $j \in J_u$ that maximises the resulting entropy H of the particle set.

By eq. (B.31), H satisfies

$$\begin{aligned}
H = & \frac{-\sum_{j \in J_d} w_t'^{[j]} \ln(w_t'^{[j]}) - \sum_{j \in J_u} w_{t-1}^{[j]} p_t(\Theta_t^{[j]}) e^{-\psi_u} \ln(w_{t-1}^{[j]} p_t(\Theta_t^{[j]}) e^{-\psi_u})}{\sum_{j \in J_d} w_t'^{[j]} + e^{-\psi_u} \sum_{j' \in J_u} w_{t-1}^{[j']} p_t(\Theta_t^{[j']})} \\
& + \ln\left(\sum_{j \in J_d} w_t'^{[j]} + e^{-\psi_u} \sum_{j' \in J_u} w_{t-1}^{[j']} p_t(\Theta_t^{[j']})\right) ,
\end{aligned} \tag{B.40}$$

where $w_t'^{[j]}$ is the unnormalised time t particle weight (defined as in eq. (3.16)) of a particle

* Whatever the true maximum value is, rescaling it to 1 makes no difference, since the weights will ultimately be normalised.

with index $j \in J_d$, $w_{t-1}^{[j']}$ is the normalised time $t - 1$ particle weight for each particle j' , and p_t is the (uniform) time t prior. For convenience, we use the following substitutions:

$$\begin{aligned} a &= \sum_{j \in J_d} w_t^{[j]} \ln \left(w_t^{[j]} \right) & b &= \sum_{j \in J_u} w_{t-1}^{[j]} p_t(\Theta_t^{[j]}) \ln \left(w_{t-1}^{[j]} p_t(\Theta_t^{[j]}) \right) \\ c &= \sum_{j \in J_d} w_t^{[j]} & d &= \sum_{j \in J_u} w_{t-1}^{[j]} p_t(\Theta_t^{[j]}) , \end{aligned} \quad (\text{B.41})$$

from which we get

$$H = \frac{-a - be^{-\psi_u} + d\psi_u e^{-\psi_u}}{c + de^{-\psi_u}} + \ln(c + de^{-\psi_u}) . \quad (\text{B.42})$$

By eq. (B.32), differentiating w.r.t. ψ_u gives

$$\begin{aligned} \frac{dH}{d\psi_u} &= \frac{1}{c + de^{-\psi_u}} \left(be^{-\psi_u} - d\psi_u e^{-\psi_u} + de^{-\psi_u} - de^{-\psi_u} \left(\frac{a + be^{-\psi_u} - d\psi_u e^{-\psi_u}}{c + de^{-\psi_u}} + 1 \right) \right) \\ &= \frac{e^{-\psi_u} \left((be^{-\psi_u} - d\psi_u e^{-\psi_u} + de^{-\psi_u}) (c + de^{-\psi_u}) - d(a + be^{-\psi_u} - d\psi_u e^{-\psi_u} + c + de^{-\psi_u}) \right)}{(c + de^{-\psi_u})^2} \\ &= \frac{e^{-\psi_u} (bc - dc\psi_u - da)}{(c + de^{-\psi_u})^2} . \end{aligned} \quad (\text{B.43})$$

Equating the derivative to 0 then gives

$$\frac{dH}{d\psi_u} = 0 \Rightarrow \psi_u = \frac{bc - ad}{dc} . \quad (\text{B.44})$$

Also, it is clear from the last line of eq. (B.43) that

$$\psi_u > \frac{bc - ad}{dc} \Rightarrow \frac{dH}{d\psi_u} < 0 , \quad (\text{B.45})$$

so the non-negative patch likelihood energy that maximises H always satisfies

$$\psi_u = \max \left(0, \frac{b}{d} - \frac{a}{c} \right) . \quad (\text{B.46})$$

Appendix C

Algorithms

Function `PropagateParticles` (page 248) and its helpers below are a pseudocode implementation of the whole importance sampling algorithm that we proposed in chapter 3.

Function `CalcDeltaEMD` (page 254) and its helper below are a pseudocode implementation of the dissimilarity function that we proposed in chapter 4, that calculates the Earth Mover's Distance between a weighted histogram of squared pixel differences and the Dirac delta function.

Function PropagateParticles($\Theta_{t-1}^{[1:n]}, w_{t-1}^{[1:n]}$)

Given the time $t - 1$ particle set, this function samples the time t particle states.

Input : The time $t - 1$ particles states $\Theta_{t-1}^{[1:n]}$ and weights $w_{t-1}^{[1:n]}$.

Output: The time t particles states $\Theta_t^{[1:n]}$.

 $(H, G) := \text{CalcGroups}(\Theta_{t-1}^{[1:n]}, w_{t-1}^{[1:n]})$
// $\hat{\Psi}$ will store the cached discrete, bounded-displacement energy functions.
 $\hat{\Psi} := \emptyset$
// \hat{D} will store the cached displacement sampling distributions.
 $\hat{D} := \emptyset$
foreach Group key k of H **do**
if $\neg \text{ContainsKey}(\hat{D}, k)$ **then**
// Create and cache displacement sampling distribution.
foreach Grid point $\mathbf{g} \in G[k]$ **do**
if $\neg \text{ContainsKey}(\hat{\Psi}, \mathbf{g})$ **then**
// Create and cache discrete, bounded-displacement energy function.
 $\psi_t(\cdot; \mathbf{g}) := \text{discrete, bounded-displacement energy function, constructed as in eq. (3.23).}$
 $\hat{\Psi}[\mathbf{g}] := \psi_t(\cdot; \mathbf{g})$
end
end
 $\hat{D}[k] := \text{CalcDispDistr}(G[k], \hat{\Psi})$
end
// Propagate particles
foreach $(i, \Theta, w) \in H[k]$ **do**
// NOTE: each ref. to \mathbf{U} denotes a new instance of a uniform random
// variable sampled over $[0, 1)$.
 $\Theta_t^{[i]} := \Theta$
 $\mathbf{d}' \sim \hat{D}[k]$
 $\Theta_{t, [d_x, d_y]}^{[i]} += \mathbf{d}' + (\mathbf{U} - 0.5, \mathbf{U} - 0.5)^T \delta_d$
 $f := \text{CalcRotDistr}(G[k], \hat{\Psi}, \Theta_t^{[i]})$
 $\phi' \sim f$
 $\Theta_{t, [\phi]}^{[i]} += \phi' + (\mathbf{U} - 0.5) \delta_\phi$
 $\kappa' \sim \text{CalcSkewDistr}(G[k], \hat{\Psi}, \Theta_t^{[i]})$
 $\Theta_{t, [\kappa_x, \kappa_y]}^{[i]} += \kappa' + ((\mathbf{U} - 0.5) \delta_{\kappa_x}, (\mathbf{U} - 0.5) \delta_{\kappa_y})^T$
 $\mathbf{S}' \sim \text{CalcScaleDistr}(G[k], \hat{\Psi}, \Theta_t^{[i]})$
 $\Theta_{t, [s_x, s_y]}^{[i]} := \text{diag}(\delta_{s_x}^{\mathbf{U}-0.5}, \delta_{s_y}^{\mathbf{U}-0.5}) \mathbf{S}' \Theta_{t, [s_x, s_y]}^{[i]}$
end
return $\Theta_t^{[1:n]}$
end

Function CalcGroups($\Theta_{t-1}^{[1:n]}, w_{t-1}^{[1:n]}$)

This function calculates particle hash keys and groups them together.

Input : The time $t - 1$ particles states $\Theta_{t-1}^{[1:n]}$ and weights $w_{t-1}^{[1:n]}$.

Output: The particle groups as a hash table H , and the group grid points as a hash table G .

// Group particles and store the groups in hash table H .

$H := \emptyset$

for $i := 1$ **to** n **do**

$k :=$ key defined by grid squares containing particle patch vertices $\mathbf{V}(\Theta_{t-1}^{[i]})$.

$H[k] := H[k] \cup \{(i, \Theta_{t-1}^{[i]}, w_{t-1}^{[i]})\}$

end

// Calculate grid points for each group and store them in hash table G .

$G := \emptyset$

foreach Key k of H **do**

$\bar{\mathbf{A}} := \mathbf{0}$

$\bar{\mathbf{d}} := \mathbf{0}$

$W := 0$

// Average group particles.

foreach $(\cdot, \Theta, w) \in H[k]$ **do**

$\bar{\mathbf{A}} += w \log(\mathbf{A}(\Theta))$

$\bar{\mathbf{d}} += w \mathbf{d}(\Theta)$

$W += w$

end

$\bar{\mathbf{A}} := \exp\left(\frac{\bar{\mathbf{A}}}{W}\right)$

$\bar{\mathbf{d}} := \frac{\bar{\mathbf{d}}}{W}$

$\theta :=$ transformation parameters defined by $\bar{\mathbf{d}}$ and RSK decomposition of $\bar{\mathbf{A}}$.

$G[k] :=$ set of grid points contained in patch $\mathbf{V}(\theta)$.

end

return (H, G)

Function CalcDispDistr($G, \hat{\Psi}$)

This function combines discrete, bounded-displacement energy functions to construct a displacement sampling distribution.

Input : A set of grid points G and a hash table $\hat{\Psi}$ of the discrete, bounded-displacement energy functions for each point.

Output: A displacement sampling distribution f over domain D , defined as in eq. (3.22).

$Y := \text{InitDistr}(D)$

foreach *Grid point* $g \in G$ **do**

$l := \text{true}$ if and only if g is the last element of G .

foreach $d' \in D$ **do**

// Add energy $\hat{\Psi}[g](d')$ to those so far accumulated for state d' .

$Y := \text{UpdateEnergy}(\hat{\Psi}[g](d'), d', Y, l)$

end

end

$f := \text{ConvEnergiesToProbs}(Y, D)$

return f

Function InitDistr(X)

This function initialises and returns a structure of temporary variables used in the calculation of the sampling distributions.

Input : A finite, discrete domain X .

Output: A structure Y of initialised temporary variables.

// $Y.\psi$ is a map from X to \mathbb{R} that accumulates the sampling distribution energies

// for each $x \in X$.

$Y.\psi(X) := 0$

// $Y.\lambda$ is a map from X to \mathbb{N} that accumulates the number of defined energy

// terms for each $x \in X$.

$Y.\lambda(X) := 0$

// $Y.n_{\perp}$ is used to count the number of states in X with no defined energies.

$Y.n_{\perp} := 0$

// $Y.N_{\top}$ is used to count the total number of defined energies, over all states.

$Y.N_{\top} := 0$

// $Y.\bar{\psi}^{\downarrow}$ is used to keep track of the smallest mean energy so far assigned to a

// state in X .

$Y.\bar{\psi}^{\downarrow} := \infty$

return Y

Function CalcRotDistr($G, \hat{\Psi}, \Theta$)

This function combines discrete, bounded-displacement energy functions to construct a rotation sampling distribution.

Input : A set of grid points G , a hash table $\hat{\Psi}$ of the discrete, bounded-displacement energy functions for each point, and a particle state Θ .

Output: A rotation sampling distribution f over domain \hat{R} , defined as in eq. (3.29).

$Y := \text{InitDistr}(\hat{R})$

foreach *Grid point* $\mathbf{g} \in G$ **do**

$l := \text{true}$ if and only if \mathbf{g} is the last element of G .

foreach $\phi' \in \hat{R}$ **do**

$\Theta' := \Theta$

$\Theta'_{[\phi]} += \phi'$

$\mathbf{d} := \mathcal{M}(\mathbf{g}; \Theta, \Theta') - \mathbf{g}$

$\psi := \text{result of bilinearly interpolating } \hat{\Psi}[\mathbf{g}] \text{ at } \mathbf{d}, \text{ or } \perp \text{ if } \mathbf{d} \notin D$.

// Add energy ψ to those so far accumulated for state ϕ' .

$Y := \text{UpdateEnergy}(\psi, \phi', Y, l)$

end

end

$f := \text{ConvEnergiesToProbs}(Y, \hat{R})$

return f

Function CalcSkewDistr($G, \hat{\Psi}, \Theta$)

This function combines discrete, bounded-displacement energy functions to construct a skew sampling distribution.

Input : A set of grid points G , a hash table $\hat{\Psi}$ of the discrete, bounded-displacement energy functions for each point, and a particle state Θ .

Output: A skew sampling distribution f over domain \hat{K} , defined as in eq. (3.31).

// This function is almost identical to CalcRotDistr. The only difference is that \hat{K}

// is used instead of \hat{R} , and the line in which $\Theta'_{[\phi]}$ is updated should be replaced with:

// $\Theta'_{[\kappa_x, \kappa_y]} += \kappa'$,

// where $\kappa' \in \hat{K}$ is the inner iteration variable.

Function CalcScaleDistr($G, \hat{\Psi}, \Theta$)

This function combines discrete, bounded-displacement energy functions to construct a scale sampling distribution.

Input : A set of grid points G , a hash table $\hat{\Psi}$ of the discrete, bounded-displacement energy functions for each point, and a particle state Θ .

Output: A scale sampling distribution f over domain \hat{S} , defined as in eq. (3.32).

// This function is almost identical to CalcRotDistr. The only difference is that \hat{S}

// is used instead of \hat{R} , and the line in which $\Theta'_{[\phi]}$ is updated should be replaced with:

// $\Theta'_{[s_x, s_y]} := \mathbf{S}'\Theta'_{[s_x, s_y]}$,

// where $\mathbf{S}' \in \hat{S}$ is the inner iteration variable.

Function UpdateEnergy(ψ, x, Y, l)

This function updates the accumulated energies for a sampling distribution.

Input : A new energy value ψ to be added to the energies for state x of a sampling distribution's domain, a structure Y of temporary variables created by the InitDistr function, and a boolean l which is true if and only if ψ is the last energy term for x .

Output: The updated temporary variable structure Y .

```

if  $\psi \neq \perp$  then
  //  $\psi$  is defined.
   $Y.\lambda(x)++$ 
   $Y.\psi(x) += \psi$ 
end
else if  $l \wedge Y.\lambda(x) = 0$  then
  //  $x$  has no defined energies.
   $Y.\psi(x) := \perp$ 
   $Y.n_{\perp}++$ 
end
if  $l \wedge Y.\lambda(x) > 0$  then
  // All of  $x$ 's energies have been accumulated.
   $Y.N_{\top} += Y.\lambda(x)$ 
   $Y.\psi(x) := \frac{Y.\psi(x)}{Y.\lambda(x)}$ 
  if  $Y.\psi(x) < Y.\bar{\psi}^{\downarrow}$  then
    //  $x$  has minimal mean energy.
     $Y.\bar{\psi}^{\downarrow} := Y.\psi(x)$ 
  end
end
return  $Y$ 

```

Function ConvEnergiesToProbs(Y, X)

This function converts the accumulated energies into a sampling distribution.

Input : A structure Y with a field $Y.\psi$ that maps domain X to accumulated energies.

Output: A sampling function f over domain X defined in terms of the accumulated energies.

// \bar{n} is the mean number of defined energies per state of X with at least one

// defined energy.

$\bar{n} := \frac{Y.N_{\top}}{|X| - Y.n_{\perp}}$

// ρ' stores the entropy of the unnormalised sampling distribution.

$\rho' := 0$

// F stores f 's normalising constant.

$F := 0$

foreach $x \in X$ **do**

if $Y.\psi(x) \neq \perp$ **then**

// $Y.\psi(x)$ is defined.

$f(x) := \exp\{-\bar{n}(Y.\psi(x) - Y.\bar{\psi}_{\perp})\}$

$\rho' \mathrel{+}= f(x) \ln(f(x))$

$F \mathrel{+}= f(x)$

end

end

if $Y.n_{\perp} > 0$ **then**

// Handle states with no defined energies.

if $Y.n_{\perp} < |X|$ **then**

$\psi_u := \frac{\rho'}{F}$

end

else

$\psi_u := 0$

end

$p_u := \exp\{-\psi_u\}$

foreach $x \in X$ **do**

if $Y.\psi(x) = \perp$ **then**

$f(x) := p_u$

end

end

end

$f(X) := \frac{f(X)}{F}$

return f

Function CalcDeltaEMD($\mathbf{x}_s, \mathbf{x}_t, N, p$)

This function calculates the EMD between the delta function and H_p .

Input : Lists \mathbf{x}_s and \mathbf{x}_t of pixel locations for times s and t respectively, the number of histogram bins N , and a value $p \in (0, 1]$ that determines what percentile to use as an upper bound.

Output: $\varrho = \varrho[H_p, H_\delta]$, and $h^+ = \sum_i \omega'(i)$.

$(h', h^+, d^{2\downarrow}, W) := \text{CalcUnnormHist}(\mathbf{x}_s, \mathbf{x}_t, N)$

// τ_h is the normalising factor for the bottom 100p% of h' .

$\tau_h := h^+ p$

$H'_{\hat{k}} := 0$

$H'_{\hat{k}+1} := 0$

$\Sigma_{H'_{\hat{k}}} := 0$

$\Sigma_{H'_{\hat{k}-1}} := 0$

$k := 0$

$i := 0$

$h'_{\hat{k}+1} := 0$

// Loop invariant: $H'_{\hat{k}+1} = \sum_{j=1}^i h'(j) \leq \tau_h$, $\Sigma_{H'_{\hat{k}}} = \sum_{j=1}^k H'(j)$,

// where $H'(a) \triangleq \sum_{j=1}^a h'(j)$.

while $i < N \wedge H'_{\hat{k}+1} \leq \tau_h$ **do**

$k := i$

$\Sigma_{H'_{\hat{k}-1}} := \Sigma_{H'_{\hat{k}}}$

$\Sigma_{H'_{\hat{k}}} += H'_{\hat{k}+1}$

$i ++$

$h'_{\hat{k}+1} := h'(i)$

$H'_{\hat{k}} := H'_{\hat{k}+1}$

$H'_{\hat{k}+1} += h'_{\hat{k}+1}$

end

// Post-condition: $h'_{\hat{k}+1} = h'(k+1)$, $H'_{\hat{k}} = \sum_{j=1}^k h'(j) \leq \tau_h$, $H'_{\hat{k}+1} = \sum_{j=1}^{k+1} h'(j)$,

// $p < 1 \rightarrow \tau_h < H'_{\hat{k}+1}$.

$\varrho := 2k - \frac{\Sigma_{H'_{\hat{k}-1}} + \Sigma_{H'_{\hat{k}}}}{\tau_h}$

if $p < 1$ **then**

$\varrho += \frac{\tau_h \left(1.0 - \frac{H'_{\hat{k}}}{\tau_h}\right)^2}{h_{\hat{k}+1}}$

else

 // $k = N - 1$, but its value should be N .

$\varrho += 2 - \frac{H'_{\hat{k}} + H'_{\hat{k}+1}}{\tau_h}$

end

$\varrho := d^{2\downarrow} + \frac{W}{2} \varrho$

return (ϱ, h^+)

Function CalcUnnormHist($\mathbf{x}_s, \mathbf{x}_t, N$)

This function calculates a weighted unnormalised histogram over the squared Euclidean differences between image pixels.

Input : The lists of time s pixels \mathbf{x}_s and time t pixels \mathbf{x}_t , and the number of histogram bins N .

Output: The unnormalised histogram h' , its total mass h^+ , the bin width W , and the minimum squared Euclidean difference $d^{2\downarrow}$.

$d^2(\cdot) :=$ a length $|\mathbf{x}_s|$ array of real numbers.

$d^{2\downarrow} := \infty$

$d^{2\uparrow} := -\infty$

for $i := 1$ **to** $|\mathbf{x}_s|$ **do**

$d^2(i) := d^2(\mathbf{x}_s(i), \mathbf{x}_t(i))$

if $d^2(i) < d^{2\downarrow}$ **then**

$d^{2\downarrow} := d^2(i)$

end

if $d^2(i) > d^{2\uparrow}$ **then**

$d^{2\uparrow} := d^2(i)$

end

end

$W := \frac{d^{2\uparrow} - d^{2\downarrow}}{N}$

$h'(1 : |\mathbf{x}_s|) := 0$

$h^+ := 0$

for $i := 1$ **to** $|\mathbf{x}_s|$ **do**

$j := \max\left(1, \left\lceil \frac{d^2(i) - d^{2\downarrow}}{W} \right\rceil\right)$

$\omega' := \omega'_{s,t}(\mathbf{x}_s(i), \mathbf{x}_t(i))$

$h'(j) += \omega'$

$h^+ += \omega'$

end

return $(h', h^+, d^{2\downarrow}, W)$

Bibliography

- [1] L. Adhami and E. Coste-Maniere. A versatile system for computer integrated mini-invasive robotic surgery. *Lecture Notes in Computer Science*, 2488:272–281, 2002.
- [2] Koray Ak, Gerhard Wimmer-Greinecker, Omer Dzemali, Anton Moritz, and Selami Dogan. Totally endoscopic sequential arterial coronary artery bypass grafting on the beating heart. *Canadian Journal of Cardiology*, 23(5):391–392, 2007.
- [3] Marc Alexa. Linear combination of transformations. In *Proceedings of ACM SIGGRAPH 2002*, 2002.
- [4] M. Sanjeev Arulampalam, Simon Maskell, Neil Gordon, and Tim Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50:174–188, 2002.
- [5] Simon Baker, Daniel Scharstein, J.P. Lewis, Stefan Roth, Michael J. Black, and Richard Szeliski. A database and evaluation methodology for optical flow. Technical report, Microsoft Research, 2009.
- [6] J. L. Barron, D. J. Fleet, and S. S. Beauchemin. Performance of optical flow techniques. *International Journal of Computer Vision*, 12:43–77, 1994.
- [7] Richard H. Bartels, John C. Beatty, and Brian A. Barsky. *An Introduction to Splines for use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann Publishers, Inc., 1987.

- [8] Mokhtar S. Bazaraa, Hanif D. Sherali, and C. M. Shetty. *Nonlinear Programming: Theory and Algorithms*. Wiley, 2006.
- [9] Alex N. Beavers, Jr. The matrix sign function and computations in systems. *Applied Mathematics and Computation*, 2:63–94, 1976.
- [10] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 24:509–522, 2002.
- [11] F. J. Benetti, C. Ballester, G. Sani, P. Doonstra, and J. Grandjean. Video assisted coronary bypass surgery. *Journal of Cardiac Surgery*, 10(6):620–625, 1995.
- [12] Andrew Blake and Michael Isard. *Active contours: the application of techniques from graphics, vision, control theory and statistics to visual tracking of shapes in motion*. Springer, 1998.
- [13] James F. Blinn. Models of light reflection for computer synthesized pictures. In *Proceedings of the 4th annual conference on Computer graphics and interactive techniques*, SIGGRAPH '77, pages 192–198, 1977.
- [14] Jean-Yves Bouguet. Pyramidal implementation of the lucas kanade feature tracker description of the algorithm. Intel Corporation Microprocessor Research Labs, 2000.
- [15] G. E. P. Box and Mervin E. Muller. A note on the generation of random normal deviates. *Annals of Mathematical Statistics*, 29:610–611, 1958.
- [16] Samuel R. Buss. *3-D Computer Graphics: A Mathematical Introduction with OpenGL*. Cambridge University Press, 2003.
- [17] George Casella and Roger L. Berger. *Statistical Inference*. Duxbury, 2002.
- [18] P. E. Castro, W. H. Lawton, and E. A. Sylvestre. Principal modes of variation for processes with continuous sample curves. *Technometrics*, 28(4):329–337, November 1986.

- [19] Thomas F. Coleman and Yuying Li. On the convergence of interior-reflective newton methods for nonlinear minimization subject to bounds. *Mathematical Programming*, 67:189–224, November 1994.
- [20] Thomas F. Coleman and Yuying Li. An interior trust region approach for nonlinear minimization subject to bounds. *SIAM Journal on Optimization*, 6(2):418–445, 1996.
- [21] R. L. Cook and K. E. Torrance. A reflectance model for computer graphics. *ACM Transactions on Graphics*, 1(1):7–24, 1982.
- [22] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding*, 61:38–59, 1995.
- [23] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 1998.
- [24] Walter J. Culver. On the existence and uniqueness of the real logarithm of a matrix. *Proceedings of the American Mathematical Society*, 17(5):1146–1151, October 1966.
- [25] Andrew J. Davison, Ian D. Reid, Nicholas D. Molton, and Olivier Stasse. Monoslam: Real-time single camera slam. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:1052–1067, 2007.
- [26] Carl de Boor. *A Practical Guide to Splines*. Springer, 2001.
- [27] S. Dogan, T. Aybek, E. Andressen, C. Byhahn, S. Mierdl, K. Westphal, G. Matheis, A. Moritz, and G. Wimmer-Greinecker. Totally endoscopic coronary artery bypass grafting on cardiopulmonary bypass with robotically enhanced telemanipulation: Report of forty-five cases. *Journals of Thoracic Cardiovascular Surgery*, 123:1125–1131, 2002.
- [28] Arnaud Doucet, Nando de Freitas, and Neil Gordon, editors. *Sequential Monte Carlo Methods in Practice*. Springer-Verlag, 2001.
- [29] Arnaud Doucet, Neil J. Gordon, and Vikram Krishnamurthy. Particle filters for state estimation of jump markov linear systems. *IEEE Transactions on Signal Processing*, 49:613–624, 2001.

- [30] Jean Duchon. *Splines minimizing rotation-invariant semi-norms in Sobolev spaces*. Springer Berlin / Heidelberg, 1977.
- [31] Richard Durstenfeld. Algorithm 235: Random permutation. *Communications of the ACM*, 7(7):420, 1964.
- [32] V. Falk, A. Diegeler, T. Walther, J. Banusch, J. Brucerius, J. Raumans, R. Autschbach, and F. W. Mohr. Total endoscopic computer enhanced coronary artery bypass grafting. *European Journals of Cardio-Thoracic Surgery*, 17:38–45, 2000.
- [33] V. Falk, F. Mourgues, L. Adhami, S. Jacobs, H. Thiele, S. Nitzsche, F. W. Mohr, and T. Coste-Maniere. Cardio navigation: Planning, simulation, and augmented reality in robotic assisted endoscopic bypass grafting. *Annals of Thoracic Surgery*, 79:2040–2048, 2005.
- [34] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.
- [35] Keith O. Geddes, M. Lawrence Glasser, Reg A. Moore, and Tony C. Scott. Evaluation of classes of definite integrals involving elementary functions via differentiation of special functions. *Applicable Algebra in Engineering, Communication and Computing*, pages 149–165, 1990.
- [36] Stuart Geman and Donald Geman. Stochastic relaxation, gibbs distributions and the bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.
- [37] John Geweke. Bayesian inference in econometric models using monte carlo integration. *Econometrica*, 57:1317–1339, 1989.
- [38] Federico Girosi, Michael Jones, and Tomas Poggio. Regularization theory and neural networks architectures. *Neural Computation*, 7:219–269, 1995.

- [39] David Goldberg. What every computer scientist should know about floating-point arithmetic. *ACM Computing Surveys (CSUR)*, 23(1):5–48, 1991.
- [40] Gene H. Golub and Charles F. van Loan. *Matrix Computations*. The John Hopkins University Press, 1996.
- [41] N. J. Gordon, D. J. Salmond, and A. F. M. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEE Proceedings F*, 140(2):107–113, 1993.
- [42] J. C. Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, March 1975.
- [43] John C. Gower and Garnt B. Dijksterhuis. *Procrustes Problems*. Oxford Statistical Science Series, 30. Oxford University Press, 2004.
- [44] W. E. L. Grimson. A computational theory of visual surface interpolation. *Philosophical Transactions of the Royal Society of London*, 298:395–427, 1982.
- [45] Gregory D. Hager and Peter N. Belhumeur. Efficient region tracking with parametric models of geometry and illumination. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 20(10):1025–1039, October 1998.
- [46] Peter W. Hallinan. A low-dimensional representation of human faces for arbitrary lighting conditions. In *Proceedings of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Proceedings CVPR '94, pages 995–999, 1994.
- [47] John H. Halton. A retrospective and prospective survey of the monte carlo method. *SIAM Review*, 12(1):1–63, January 1970.
- [48] Tony Heap and David Hogg. Extending the point distribution model using polar coordinates. *Image and Vision Computing*, 14:589–599, 1996.
- [49] Geoffrey Hinton. Products of experts. In *Ninth International Conference on Artificial Neural Networks*, volume 1, pages 1–6, 1999.

- [50] Mingxing Hu, Graeme Penney, Daniel Rueckert, Philip Edwards, Michael Figl, Philip Pratt, and David Hawkes. A novel algorithm for heart motion analysis based on geometric constraints. In *Medical Image Computing and Computer-Assisted Intervention MICCAI 2008*, volume 5241 of *Lecture Notes in Computer Science*, pages 720–728, 2008.
- [51] Intuitive Surgical. The da Vinci surgical system. www.intuitivesurgical.com.
- [52] Michael Isard and Andrew Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.
- [53] Edwin T. Jaynes. Information theory and statistical mechanics. *Physical Review*, 106(4):620–630, May 1957.
- [54] I.T. Jolliffe. *Principal Component Analysis*. Springer Series in Statistics. Springer, 2002.
- [55] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. *Transactions of the ASME - Journal of Basic Engineering*, 82(Series D):35–45, 1960.
- [56] U. Kappert, R. Cichon, J. Schneider, V. Gulielmos, T. Ahmadzade, J. Nicolai, S. M. Tugtekin, and S. Schueler. Technique of closed chest coronary artery surgery on the beating heart. *European Journals of Cardio-Thoracic Surgery*, 20:765–769, 2001.
- [57] Charles Kenney and Alan J. Laub. Condition estimates for matrix functions. *SIAM Journal on Matrix Analysis and Applications*, 10:191–209, 1989.
- [58] Kwang In Kim, Matthias O. Franz, and Bernhard Schölkopf. Iterative kernel principal component analysis for image modeling. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27:1351–1366, 2005.
- [59] Georg Klein and David Murray. Full-3d edge tracking with a particle filter. In *Proceedings of the British Machine Vision Conference*, BMVC06, 2006.
- [60] Augustine Kong. A note on importance sampling using standardized weights. Technical Report 348, The University of Chicago, 1992.

- [61] Augustine Kong, Jun S. Liu, and Wing Hung Wong. Sequential imputations and bayesian missing data problems. *Journal of the American Statistical Association*, 89:278–288, 1994.
- [62] D. N. Lawley and A. E. Maxwell. Factor analysis as a statistical method. *Journal of the Royal Statistical Society, Series D*, 12:209–229, 1962.
- [63] J. Leven, D. Burschka, R. Kumar, G. Zhang, S. Blumenkranz, X. T. Dai, M. Awad, G. D. Hager, M. Marohn, M. Choti, C. Hasser, and R. H. Taylor. Davinci canvas: a telerobotic surgical system with integrated, robot-assisted, laparoscopic ultrasound capability. *Lecture Notes in Computer Science*, 3749:811–818, 2005.
- [64] Elizaveta Levina and Peter Bickel. The earth mover’s distance is the mallows distance: Some insights from statistics. In *Eighth IEEE International Conference on Computer Vision (ICCV)*, volume 2, pages 251–256, 2001.
- [65] Donald Lloyd-Jones, Robert J. Adams, Todd M. Brown, Mercedes Carnethon, Shifan Dai, Giovanni De Simone, T. Bruce Ferguson, Earl Ford, Karen Furie, Cathleen Gillespie, Alan Go, Kurt Greenlund, Nancy Haase, Susan Hailpern, P. Michael Ho, Virginia Howard, Brett Kissela, Steven Kittner, Daniel Lackland, Lynda Lisabeth, Ariane Marelli, Mary M. McDermott, James Meigs, Dariush Mozaffarian, Michael Mussolino, Graham Nichol, Vronique L. Roger, Wayne Rosamond, Ralph Sacco, Paul Sorlie, Randall Stafford, Thomas Thom, Sylvia Wasserthiel-Smoller, Nathan D. Wong, Judith Wylie-Rosett, and on behalf of the American Heart Association Statistics Committee and Stroke Statistics Subcommittee. Heart disease and stroke statistics–2010 update: A report from the american heart association. *Circulation*, 121(7):e46–215, 2010.
- [66] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of Imaging Understanding Workshop*, pages 121–130, 1981.
- [67] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Image and Vision Computing*, volume 22 (10), pages 761–767, 2004.

- [68] Uwe Mehlhorn, Karen L. Davis, Edward J. Burke, Deborah Adams, Glen A. Laine, and Steven J. Allen. Impact of cardiopulmonary bypass and cardioplegic arrest on myocardial lymphatic function. *American Journal of Physiology - Heart and Circulatory Physiology*, 268:H178–H183, 1995.
- [69] Peter Mountney and Guang-Zhong Yang. Soft tissue tracking for minimally invasive surgery: Learning local deformation online. In *Medical Image Computing and Computer-Assisted Intervention MICCAI 2008*, 2008.
- [70] G. J. Murphy and G. D. Angelini. Side effects of cardiopulmonary bypass: What is the reality? *Journal of Cardiac Surgery*, 19:481–488, 2004.
- [71] Saralees Nadarajaha. A generalized normal distribution. *Journal of Applied Statistics*, 32:685–694, 2005.
- [72] Addy Ngan, Frdo Durand, and Wojciech Matusik. Experimental validation of analytical brdf models. In *ACM SIGGRAPH 2004 Sketches*, page 90, 2004.
- [73] Jorge Nocedal and Stephen Wright. *Numerical Optimization*. Springer, 2000.
- [74] Jeffrey Nunemacher. Which real matrices have real logarithms? *Mathematics Magazine*, 62(2):132–135, April 1989.
- [75] Erkki Oja. Simplified neuron model as a principal component analyzer. *Journal of Mathematical Biology*, 15:267–273, 1982.
- [76] Takayuki Okatani and Koichiro Deguchi. Shape reconstruction from an endoscope image by shape from shading technique for a point light source at the projection center. *Computer Vision and Image Understanding*, 66:119–131, 1997.
- [77] Tobias Ortmaier, Martin Groger, Dieter H. Boehm, Volkmar Falk, and Gerd Hirzinger. Motion estimation in beating heart surgery. *IEEE Transactions on Biomedical Engineering*, 52(10):1729–1740, October 2005.
- [78] Roger Penrose. A generalized inverse for matrices. *Mathematical Proceedings of the Cambridge Philosophical Society*, 51:406–413, 1955.

- [79] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, 18(6):311–317, June 1975.
- [80] E. Prados and O. Faugeras. Shape from shading: a well-posed problem? In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2 of *CVPR 2005*, pages 870–877, 2005.
- [81] Hartmut Prautzsch, Wolfgang Boehm, and Marco Paluszny. *Bézier- and B-spline techniques*. Springer-Verlag, 2000.
- [82] William H. Press, Saul A. Teukolsky, and William T. Vetterling and Brian P. Flannery. *Numerical Recipes: The Art of Scientific Computing*. Cambridge University Press, 2007.
- [83] Mark Pupilli and Andrew Calway. Real-time camera tracking using known 3d models and a particle filter. In *Proceedings of the International Conference on Pattern Recognition*, ICPR06, August 2006.
- [84] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.
- [85] J.O. Ramsay and B.W. Silverman. *Functional Data Analysis*. Springer Series in Statistics. Springer, 2005.
- [86] H. U. Rashid and P. Burger. Differential algorithm for the determination of shape from shading using a point light source. *Image and Vision Computing*, 10(2):119–127, March 1992.
- [87] Rogério Richa, Philippe Poignet, and Chao Liu. Three-dimensional motion tracking for beating heart surgery using a thin-plate spline deformable model. *The International Journal of Robotics Research*, 29(2–3):218–230, February 2010.
- [88] Sam T. Roweis and Lawrence K. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, December 2000.
- [89] Yossi Rubner, Carlo Tomasi, and Leonidas J. Guibas. The earth mover’s distance as a metric for image retrieval. *International Journal of Computer Vision*, 40(2):99–121, 2000.

- [90] Charles-Auguste Saint-Pierre, Jonathan Boisvert, Guy Grimard, and Farida Cheriet. Detection and correction of specular reflections for automatic surgical tool segmentation in thoracoscopic images. *Machine Vision and Applications*, pages 1–0, 2007.
- [91] Peter Sand and Seth Teller. Particle video: Long-range motion estimation using point trajectories. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2195–2002. IEEE, 2006.
- [92] Terence D. Sanger. Optimal unsupervised learning in a single-layer linear feedforward neural network. *Neural Networks*, 2:459–473, 1989.
- [93] János Schanda, editor. *Colorimetry – Understanding the CIE System*. John Wiley & Sons, Inc., 2007.
- [94] Peter H. Schnemann. A generalized solution of the orthogonal procrustes problem. *Psychometrika*, 31:1–10, 1966.
- [95] Bernhard Schölkopf, Alexander Smola, and Klaus-Robert Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10:1299–1319, 1998.
- [96] Steven M. Seitz, Brian Curless, James Diebel, Daniel Scharstein, and Richard Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, 2006.
- [97] Steven A. Shafer. Using color to separate reflection components. Technical Report 136, University of Rochester, 1984.
- [98] Amnon Shashua. On photometric issues in 3d visual recognition from a single 2d image. *International Journal of Computer Vision*, 21(1–2):99–122, 1997.
- [99] Jianbo Shi and Carlo Tomasi. Good features to track. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’94)*, pages p. 593–600. IEEE, June 1994.

- [100] A. F. M. Smith and A. E. Gelfand. Bayesian statistics without tears: A sampling-resampling perspective. *The American Statistician*, 46:2, 1992.
- [101] Viv Peto Sophie Petersen and Mike Rayner. *Coronary heart disease statistics 2004*. British Heart Foundation Health Promotion Research Group, 2004.
- [102] D. Stoyanov, G. P. Mylonas, F. Deligianni, A. Darzi, and G. Z. Yang. Soft-tissue motion tracking and structure estimation for robotic assisted MIS procedures. *Medical Image Computing And Computer-Assisted Intervention - MICCAI 2005, Pt 2*, 3750:139–146, 2005.
- [103] Danail Stoyanov and Guang Zhong-Yang. Removing specular reflection components for robotic assisted laparoscopic surgery. In *Proceedings of the IEEE International Conference on Image Processing*, volume 3 of *ICIP05*, pages 632–635, 2005.
- [104] Danail Valentinov Stoyanov. *Recovering 3D Structure and Motion for Applications in Robotic Laparoscopic Surgery*. PhD thesis, Imperial College London, 2005.
- [105] Richard Szeliski and Sing Bing Kang. Shape ambiguities in structure from motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(5):506–512, May 1997.
- [106] S. Szpala, M. Wierzbicki, G. Guiraudon, and T. M. Peters. Real-time fusion of endoscopic views with dynamic 3-d cardiac images: A phantom study. *IEEE Transactions on Medical Imaging*, 24:1207–1215, 2005.
- [107] Robby T. Tan and Katsushi Ikeuchi. Separating reflection components of textured surfaces using a single image. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27(2):178–193, February 2005.
- [108] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000.
- [109] George R. Terrell and David W. Scott. Variable kernel density estimation. *Annals of Statistics*, 20(3):1236–1265, 1992.

- [110] Mohan Thanikachalam, Pierluca Lombardi, Hassan Y. Tehrani, Kushagra Katariya, and Tomas A. Salerno. The history and development of direct coronary surgery without cardiopulmonary bypass. *Journal of Cardiac Surgery*, 19(6):516–519, 2004.
- [111] Michael E. Tipping and Christopher M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, pages 611–622, September 1999.
- [112] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, 1991.
- [113] Lorenzo Torresani, Aaron Hertzmann, and Chris Bregler. Learning non-rigid 3d shape from 2d motion. In *Advances in Neural Information Processing Systems*, volume 16 of *NIPS 2003*, 2004.
- [114] Grace Wahba. *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics, 1990.
- [115] Yair Weiss and Edward H. Adelson. Slow and smooth: A bayesian theory for the combination of local motion signals in human vision. Technical report, Massachusetts Institute of Technology, 1998.
- [116] Yair Weiss, Eero P. Simoncelli, and Edward H. Adelson. Motion illusions as optimal percepts. *Nature Neuroscience*, 5(6):598–604, 2002.
- [117] Eric W. Weisstein. *The CRC Concise Encyclopedia of Mathematics*. CRC Press, 1998.
- [118] Gerhard Wimmer-Greinecker, Heinz Deschka, Tayfun Aybek, Stefan Mierdl, and Selami Dogan Anton Moritz. Current status of robotically assisted coronary revascularization. *The American Journal of Surgery*, 188(1):76–82, October 2004.
- [119] Gerhard Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods*. Springer-Verlag, 2 edition, 2003.